robotron

Systemhandbuch SCP

Anleitung für den Programmierer

S Y S T E M H A N D B U C H S C P

Anleitung fuer den Programmierer

Stand 31.12.1984

VEB Robotron Bueromaschinenwerk Soemmerda

Inhaltsverzeichnis

		Seite
1.	Einleitung	4
2.	Grundlagen fuer die Programmierung	5
2.1.	Speicherkonzept	5
2.2.	Logische Geraete und ihr Aufruf	7
2.3.	Kommando-Eingabe ueber CCP	7
3.	Beschreibung der BDOS-Funktionen	9
3.1.	Vorbemerkungen	9
3.2.	Kommunikation und Aufruf	9
3.3.	Zeichenorientierte Ein- und Ausgabe	10
3.3.1.	Zuordnung der Kanaele	10
3.3.2.	Datenverkehr ueber die Konsole	11
3.3.3.	Zeichenweiser Datenverkehr ueber die sequentiellen und Listkanaele	16
3.4.	Arbeit mit Diskettendateien	17
3.4.1.	Aufbau des FCB	17
3.4.2.	Verwaltung der Verzeichniseintragungen	18
3.4.3.	Adressierung und Datentransport	23
3.4.4.	System und Hilfsfunktionen	29
3.4.4.1.	Systemfunktionen	29
3.4.4.2.	Laufwerksfunktionen	32
3.4.4.3.	Dateienverwaltung	35
3.5.	Zusammenfassung der BDOS-Funktionen	37
3.5.1.	Uebersicht nach Funktionsgruppen geordnet	37
3.5.2.	Uebersicht nach Auftragsnummern geordnet	39
3.5.3.	Erlaeuterungen und Kommentare	41
4.	Beispiel fuer die Anwendung der BDOS-Funktionen	43
5.	Beschreibung des BIOS	51
5.1.	Aufgabe des BIOS	51
5.2.	BIOS-Schnittstelle	52
5.3.	BIOS-Bestandteile	53
5.3.1.	Initialisierung des SCP	53
5.3.2.	Logische Ein-/Ausgabekanaele	54
5.3.3.	Diskettenarbeit	56
5.3.3.1.	Laufwerksteuerung	56
5.3.3.2.	Datenverkehr	57
5.4.	BIOS-Eintrittspunkte	57
5.5.	Beschreibung der Disketten	63
5.5.1.	Diskettenparameterkopf	64
5.5.2.	Diskettenparameterblock	65

*** Inhaltsverzeichnis ***

Anhang A Reservierte Speicherplaetze

Anhang B Steuerzeichen fuer Bildschirm

Anhang C BDOS - Funktionen

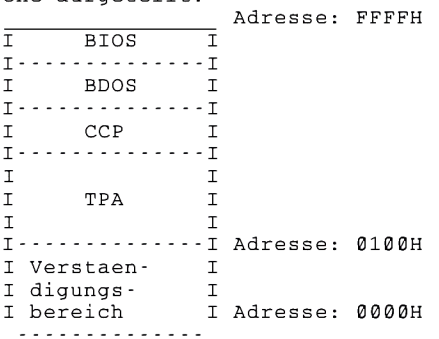
1. Einleitung

Das Erarbeiten von Anwenderprogrammen fuer ein SCP-Svstem hat vor allem dann Bedeutung, wenn nicht auf das breite Spektrum von Standardsoftware zurueckgegriffen werden kann, die Standardsoftware zu modifizieren (installieren) ist oder Sonderprobleme zu loesen sind. Voraussetzung fuer den Programmierer sind entsprechende Hardware-Kenntnisse. Diese vorliegende Beschreibung beinhaltet die Grundlagen fuer eine Programmerarbeitung. Es werden die zur Verfuegung stehenden BDOS-Funktionen und der hardwareabhaengige Teil des SCP - das BIOS - beschrieben.

2. Grundlagen fuer die Programmierung

2.1. Speicherkonzept

Nach dem Laden des Programmes SCPX von der Systemdiskette ("Kaltstart") ist der zur Verfuegung stehende 64 KByte-Speicher in 5 Bereiche aufgeteilt.



BIOS: Basic-I/O-System
Dieser Systembereich enthaelt die hardwarespezifischen Programmteile. Am Beginn steht eine Sprungtabelle, die zu den einzelnen Routinen zur Bedienung der im System eingebundenen physischen E/A-Geraete fuehrt.

BDOS: Basic Disk Operating System
Dieser Teil des Systems bietet dem Programmierer alle die Routinen an, die den problemlosen Datenaustausch mit den logischen Geraeten (siehe Pkt. 2.2) gewaehrleisten. Zur Nutzung dieser Routinen, auch BDOS-Funktionen genannt, gibt es ein standardisiertes Verfahren zu deren Aufruf (siehe Pkt. 3.2.).
BDOS bedient sich der physischen BIOS-Routinen.

CCP: Consol Command Processor
Nach "Kaltstart" bzw. Systemneustart (siehe Pkt. 3.4.4.) uebernimmt CCP die Steuerung des Betriebssystems. In einem Systemteil sind die residenten Kommandos DIR, REN, TYPE, ERA, SAVE und USER realisiert, die den TPA nicht veraendern. (Diese Kommandos sind in der "Anleitung fuer den Bediener" umfassend beschrieben.)
Ausserdem koennen mit CCP auch die transienten Funktionen aufgerufen werden. Diese Programme werden in den TPA geladen. Dann uebergibt CCP die Steuerung an die transienten Programme. Nach Abarbeitung wird die Steuerung an das CCP-Programm zurueckgegeben

TPA: Transient Program Area
Dies ist der Speicherbereich, der den transienten Kommandos und Programmen zur Verfuegung steht und bei der Adresse 0100H beginnt. Ab dieser Adresse werden die transienten Kommandos oder Programme geladen, die dann auch bei 0100H gestartet werden.
Dieser Bereich steht ebenfalls fuer den Anwenderprogrammierer zur Verfuegung.
Die Groesse des TPA wird in der Kaltstartmeldung auf dem Bildschirm angezeigt.

Verstaendigungsbereich:
In diesem Bereich sind Adressen, Kennbytes, Spruenge und Standardpuffer fuer die Arbeit des SCPX abgelegt:

Adresse	Bedeutung
0000H - 0002H	Sprung an den Sprungbefehl fuer Warmstart in der BIOS-Sprungtabelle. (Dies ist der zweite Sprung in dieser Tabelle.)
0004H	aktuelles logisches Laufwerk z.B.: - LW A: Inhalt der Adresse 0004H . = 0 . - LW D: Inhalt der Adresse 0004H = 3
0005H - 0007H	BDOS-Ruf.Die Adresse auf 0006H und 0007H - also die Anfangsadresse von BDOS - kennzeichnet auch die obere Grenze des TPA
0008H - 003AH	reserviert fuer Restart-Anspruenge in transienten Programmen. (siehe Pkt. 8.).
003EH - 004FH	reserviert
005CH - 007CH	Standard-Dateisteuerblock file control block - FCB (siehe auch Pkt. 2.3.).
0080H - 00FFH	Standard Ein-/Ausgabe-Puffer (direct memory access - DMA (siehe auch Pkt. 2.3.).

2.2. logische Geraete und ihr Aufruf

Vom SCPX werden folgende logische Geraete unterstuetzt:

- die Konsole fuer Dialogverkehr

- ein Listgeraet

- 16 logische FD-Laufwerke mit der Bezeichnung A...P als Datentraeger

- Ausgabe und Eingabe ueber sequentielle Datenkanaele

Die Konsole enthaelt fuer die Zeicheneingabe eine Tastatur, zur Zeichenausgabe einen Bildschirm mit dem Format 80x24 Zeichen oder 64x16 Zeichen.

Um mit den logischen Geraeten arbeiten zu koennen, sind die BDOS-bzw. BIOS-Rufe zu verwenden (siehe Pkt. 3.).

2.3. Kommando-Eingabe ueber CCP

Ein Kommando-/Programmaufruf erfolgt stets durch die Eingabe des residenten Kommandos oder des Dateinamens des transienten Programmes. Der Dateityp COM des transienten Programmes darf nicht eingegeben werden. Danach ist noch die Eingabe von Parametern moeglich. Diese werden generell auf den Adressen 0081H bis 00FBH abgelegt. Es sind somit insgesamt 123 Zeichen bei der Parametereingabe moeglich, die tatsaechliche Anzahl eingegebener Zeichen wird auf der Adresse 0080H abgelegt.
Der Programmierer hat aber darauf zu achten, dass die Adresse 0080H nach einem Kalt- und Warmstart die aktuelle DMA-Adresse fuer Floppy-Disk-Zugriffe ist. Werden die Parameter noch nach einem FD-Zugriff benoetigt, muss der Programmierer diese Parameter in einen anderen Speicherbereich verschieben oder eine andere DMA-Adresse definieren (siehe Pkt.3., Funktion 26).

Neben der Speicherung der Parameter ab 0080H werden ausserdem die direkt dem Kommando bzw. Programmnamen moeglicherweise folgenden zwei Dateibezeichnungen (Laufwerkscode, Dateiname, Dateityp) auf 005CH und 006CH abgelegt. Sind der Name und Typ der Datei kuerzer als 11 Zeichen, werden die freien Bytes mit 20H belegt. Zwischen Dateiname und -typ ist bei der Eingabe ein Punkt einzugeben.
Die Adresse 005CH ist aber auch als Standardadresse fuer den FCB nach Kalt- und Warmstart definiert. Dies hat der Programmierer in gleicher Weise zu beachten wie die Nutzung des Speicherbereichs ab 0080H.
Bei der Speicherung der Parameter ab 005CH, 006CH und 0080H werden generell alle eingegebenen Buchstaben als Grossbuchstaben abgelegt!

Beispiel:

A>TEST 123.abc B:5.DE fgh<ET>

Es wird gespeichert:

- auf 005CH: 00H
  auf 005DH: 31H ("1")
  auf 005EH: 32H ("2")
  auf 005FH: 33H ("3")
  auf 0060H
  bis 0064H: 20H
  auf 0065H: 41H ("A")
  auf 0066H: 42H ("B")
  auf 0067H: 43H ("C")

- auf 006CH: 02H
  auf 006DH: 35H ("5")
  auf 006EH
  bis 0074H: 20H
  auf 0075H: 44H ("D")
  auf 0076H: 45H ("E")
  auf 0077H: 20H

- auf 0080H bis 0093H:
  13H, 20H, 31H, 32H, 33H, 2EH, 41H, 42H, 43H, 20H, 42H,
  3AH, 35H, 2EH, 44H, 45H, 20H, 46H, 47H, 48H

Ein Kommando- oder Programmaufruf erfolgt stets vom CCP—Status aus, d.h. das Kommando bzw. Programm kann mit RET zu CCP zurueckkehren. Der Programmierer muss aber unbedingt beachten:

— Der von CCP bereitgestellte Stack ist nur 16 Byte gross, wobei bereits die adresshoechsten Bytes die Rueckkehradresse zu CCP enthalten. Deshalb sollte in jedem Programm ein eigener Stack definiert werden. Vor dem RET zur Rueckkehr in das CCP ist dann der alte Stackpointer wieder einzustellen.

— Anderenfalls ist das Programm unbedingt mit dem System-Neustart (siehe Pkt.3.4.4.1.) zu beenden, damit CCP von der Systemdiskette im Laufwerk A neu geladen wird. Dadurch wird der CCP-Status wieder erreicht, ohne dass der alte Stackpointerstand eingestellt werden muss.
  Es ist zweckmaessig, das Anlegen eines programmeigenen Stacks und das Beenden eines Programms mit dem System-Neustart generell vorzusehen.

3. Beschreibung der BDOS—Funktionen

3.1. Vorbemerkungen

Das Diskettengrundbetriebssystem (BDOS) stellt den Kern des SCPX-Betriebssystems dar. Es unterstuetzt die Ein-und Ausgabe auf den logischen Geraeten (Konsole, Listgeraet, sequ. Datenkanaele), sowie die Dateiarbeit auf den Disketten.
Ueber BDOS-Ruf koennen folgende Auftraege von BDOS angefordert werden:

- Zeichenorientierte Ein-und Ausgabe

  -Zuordnung der logischen zu den physischen Ein-und Ausgabekanaelen
  -zeichenweise Ein-und Ausgabe ueber die Konsole
  -zeichenweise Ein-und Ausgabe ueber die sequentiellen Datenkanaele
  -zeichenweise Ausgabe ueber den Listkanal

- Arbeit mit Diskettendateien

  -allgemeine Dateimanipulationen und -verwaltung
  -sequentieller Dateizugriff
  -direkter Dateizugriff

-Systemverwaltung

  -Initialisierung
  -Ermittlung des Systemzustandes
  -weitere Hilfsfunktionen

3.2. Kommunikation und Aufruf

Der BDOS-Ruf erfolgt ueber einen UP-Aufruf der absoluten Speicher-Adresse 5. Dieser Speicherbereich liegt im Verstaendigungsbereioh. An dieser Stelle befindet sich ein Sprungbefehl, der dort bei der Systeminitialisierung eingetragen wird. Dieser Sprungbefehl setzt im BDOS-Koerper auf.Die Rueckkehr aus dem BDOS wird dort selbst durch einen RET-Befehl realisiert. Der BDOS-Ruf muss noch durch Belegung von Registern spezifiert werden.
Die BDOS-Funktionen sind mit laufenden Nummern versehen. Im CPU-Register C muss vor Ausfuehrung des Rufes diese Kodenummer eingestellt werden, die der gewuenschten BDOS-Funktion zugeordnet ist.

Die zusaetzlichen Eingabeparameter werden bei 8-Bit-Werten in das CPU-Register E, bei 16-Bit-Werten in das CPU-Doppelregister DE eingetragen. Nach Ausfuehrung der BDOS-Funktion wird das rufende Programm hinter der Aufrufstelle fortgesetzt. Ausgabeparameter nach Abarbeitung der BDOS-Funktion werden als

- 8-Bit-Werte im CPU-Register A
- 16-Bit-Werte im CPU-Doppelregister HL
- Feldeintragungen in einem Feld, das durch CPU-Doppelregister DE vor Aufruf adressiert wurde,

bereitgestellt.
Das Betriebssystem benutzt einen eigenen Stapel, so dass das aufrufende Programm nur mit einer Stapel-Ebene durch den Aufruf belastet wird.

3.3. Zeichenorientierte Ein-und Ausgabe

3.3.1. Zuordnung der Kanaele

Der zeichenorientierte Datenaustausch mit den Peripheriegeraeten erfolgt ueber die 4 logischen Kanaele:

- Konsole
- sequentielle Eingabe
- sequentielle Ausgabe
- Listgeraet.

Jedem dieser Kanaele kann ein von jeweils 4 kanalspezifischen physischen Subkanaelen zugeordnet werden.Dieses Zuordnungsbyte (= I/O Byte) befindet sich im Verstaendigungsbereich. Die Zuordnung der Subkanaele geschieht in folgender Weise:

```
BIT 0 und 1 : Konsole
BIT 2 und 3 : sequentielle Eingabe
BIT 4 und 5 : sequentielle Ausgabe
BIT 6 und 7 : Listgeraet
```

Funktion 7: Abfrage des I/O-Bytes
Aufrufparameter:
Register C: 07H
Rueckgabewert:
Register A: IOBYTE-Wert

Bereitstellung des aktuellen Wertes des I/O-Bytes im CPU-Register A

```
I                                          I
I  Funktion 8: Setzen  des I/O-Bytes       I
I                                          I
I   Aufrufparameter:                       I
I         Register  C:   08H               I
I         Register  E:   IOBYTE-Wert       I
I                                          I
```

Die Belegung des CPU-Registers E wird in das I/O-Byte eingetragen.

3.3.2. Datenverkehr ueber die Konsole

Der Dialogverkehr zwischen Geraet und Bediener erfolgt ueber die Konsole.

Es sind die 2 Betriebsarten

- zeichenweise Ein-/Ausgabe
- gepufferte Ein-/Ausgabe

moeglich.
Der zeichenweise Datenverkehr kann entweder im BDOS-gesteuerten oder im direkten Modus erfolgen. Im ersten Modus erfolgt bei der Eingabe der Zeichen eine Echofunktion auf der Konsole, sowie eine Sonderbehandlung folgender Eingabebelegungen:

^I: Tabulation auf den naechsten 8-Spalten-Tabulator

^S: Die naechste Konsoleausgabe wird suspendiert, bis ein beliebiges Zeichen von der Konsole eingegeben wird.
Der BDOS-Ruf wird im Wartezustand nicht verlassen, d.h. das rufende Programm nicht fortgesetzt.

^P: Die nachfolgenden Ausgaben von Konsolezeichen erfolgen ueber die Kanaele Konsole und Listgeraet parallel. Durch Eingabe von ^P wird diese Funktion wieder zurueckgesetzt (triggernde Funktion).

```
I                                          I
I  Funktion 1: Konsolen - Eingabe          I
I                                          I
I   Aufrufparameter:                       I
I         Register  C:   01H               I
I                                          I
I   Rueckgabewert:                         I
I         Register  A:   Zeichen           I
I                                          I
```

Das naechste Zeichen von der Konsole wird in das Register A uebergeben. In der aufgerufenen Routine wird solange verblieben, bis das naechste Zeichen von der Konsole zur Verfuegung gestellt werden kann. Die eingegebenen Zeichen werden als Echo auf der Konsole ausgegeben.
Die Steuerzeichen ^S, ^P und ^I werden ausgewertet; ^C wird nicht ausgefiltert.

Funktion 2: Konsolen - Ausgabe
Aufrufparameter:
Register C: 02H
Register E: Zeichen

Die Belegung des CPU-Registers E wird auf der Konsole ausgegeben. Die Steuerzeichen ^S und ^P als Ausgabezeichen werden ignoriert, ^I wird beruecksichtigt. Die ueber die Konsole eingegebenen Zeichen ^S und ^P, werden bei der Ausgabe interpretiert.

Funktion 11: Konsolen - Status
Aufrufparameter:
Register C: 0BH
Rueckgabewert:
Register A: Status

Die Funktion prueft, ob bereits das naechste Zeichen auf der Konsole eingegeben worden ist. Die Belegung des CPU-Registers A bei Rueckkehr ist mit
- A = 00H als keine Zeicheneingabe und
- A = 01H als erfolgte Zeicheneingabe

zu interpretieren. Das eingegebene Zeichen verursacht solange den Belegt-Status, bis das Zeichen ueber Konsolen-Eingabe abgerufen wird.

Funktion 6: Konsolenein- und ausgabe direkt
Aufrufparameter:
Register C: 06H
Register E: 0FFH (Eingabe) oder Zeichen (Ausgabe)
Rueckgabewert:
Register A: Zeichen oder Status

Enthaelt das CPU-Register E den Wert 0FFH, so erfolgt eine Konsoleneingabe in das CPU-Register A. Wurde kein Zeichen ueber die Konsole eingegeben, wird das CPU-Register A mit dem Kode 00H belegt. Es erfolgt keine Pufferung der Eingabedaten, sondern eine kombinierte Eingabe-/Status-Funktion wird realisiert.
Enthaelt das CPU-Register E einen von 0FFH verschiedenen Wert, wird dieses Zeichen auf der Konsole ausgegeben.

Funktion 9: Ausgabe einer Zeichen— kette auf Konsole
Aufrufparameter:
Register C: 09H
Register DE: Adresse der Zeichenkette

Der Inhalt des CPU-Registers DE wird als Adresse auf der Zeichenkette interpretiert. Die Zeichenkette kann beliebige Zeichen enthalten. Das erste auftretende Zeichen ¤ wird als Endekennzeichen der Zeichenkette interpretiert und nicht mit ausgegeben. BDOS-Zustaende, die ueber ^S, ^P oder ^I gesteuert werden, werden beruecksiohtigt.

```
I                                          I
I  Funktion 10: Eingabe in den Konsol-     I
I               puffer                     I
I                                          I
I   Aufrufparameter:                       I
I         Register  C:   0CH               I
I         Register DE:   Pufferadresse     I
I                                          I
I   Rueckgabewert:                         I
I         Zeichen von Konsole im Puffer    I
I                                          I
```

Diese Funktion dient der Eingabe einer ganzen Zeile in einen durch das Registerpaar DE adressierten Puffer, wobei gleichzeitig die Anzeige der eingegebenen Zeichen auf der Konsole erfolgt.
Das 1. Byte des Puffers muss die maximal eingebbare Zeichenzahl enthalten (1-255).
Das 2. Byte des Puffers enthaelt nach Rueckehr die tatsaechliche Anzahl eingegebener Zeichen und definiert den Gueltigkeitsbereich des Puffers.
Ab der 3. Position des Puffers stehen die eingegebenen Zeichen.
Der Puffer wird vor der Eingabe nicht geloescht, so dass nach dem zuletzt eingegebenen Zeichen beliebige Zeichen folgen.

Die Zeichenausgabe wird beendet durch

- Erreichen der Maximalanzahl (1.Byte des Puffers),
- Eingabe, Abschlusstaste
- Eingabe ^J (siehe Anlage D),
- Eingabe ^M (siehe Anlage D).

Der Kursor der Konsole wird danach auf den Anfang der laufenden Konsolezeile positioniert.

Eine Reihe von Steuerzeichen ermoeglichen ein einfaches zeilen orientiertes Editieren:

DEL : Das letzte Zeichen im Puffer wird geloescht, auf der Konsole aber erneut als Echo ausgegeben.

^R : Die Zeile wird nochmals ausgegeben - so, wie sie im Puffer steht. Dies ist z.B. nach DEL sinnvoll.

^C : Wird dieses Zeichen am Pufferanfang eingeschrieben, so wird ein Warmstart ausgeloest. An anderen Positionen bewirkt dieses Steuerzeichen die Ablage des Codes (03H) im Puffer und die Anzeige^C auf der Konsole.

^E : Dieses Steuerzeichen gelangt nicht in den Puffer, es bewirkt auf der Konsole eine Zeilenschaltung und eine Positionierung des Kursors an den Zeilenanfang (kein Eingabeende!).

^H : Das zuletzt eingegebene Zeichen wird sowohl im Puffer als auch auf der Konsole geloescht.

^U : Der Puffer wird geloescht. Auf der Konsole wird ein "#" ausgegeben, das Eingabeecho wird auf der naechsten Zeile angezeigt.

^X : Der Puffer wird geloescht, ebenfalls alle eingegebenen Zeichen auf der Konsole. Der Kursor nimmt die Position ein, die er vor dem Funktionsaufruf hatte.

^I : In den Puffer wird der Code (09H) eingetragen. Auf der Konsole werden so viele Leerzeichen ausgegeben, dass eine 8-Zeichen-Tabulation entsteht (Spalten 1,9,17,25...).

^P : Durch dieses Steuerzeichen wird bewirkt, dass jedes Zeichen nicht nur zur Konsole sondern auch zum Listgeraet ausgegeben wird. Diese Parallelausgabe bleibt ueber die BDOS-Funktion #10 hinaus erhalten, so dass bei folgenden Funktionen #2, #9 und #10 eine Ausgabe auf Konsole und Listgeraet erfolgt.
Die zusaetzliche Ausgabe auf das Listgeraet kann durch eine erneute Eingabe von ^P innerhalb der BDOS-Funktion #10 abgeschaltet werden.

3.3.3. Zeichenweiser Datenverkehr ueber die sequentiellen und Listkanaele

```
I                                          I
I  Funktion 3: sequentielle Eingabe        I
I                                          I
I   Aufrufparameter:                       I
I         Register  C:   03H               I
I                                          I
I   Rueckgabewert:                         I
I         Register  A:   Zeichen           I
I                                          I
```

BDOS wartet auf die Eingabe eines Zeichens ueber den sequentiellen Kanal und uebergibt das Zeichen im CPU-Register A.

```
I                                          I
I  Funktion 4: sequentielle Ausgabe        I
I                                          I
I   Aufrufparameter:                       I
I         Register  C:   04H               I
I         Register  E:   Zeichen           I
I                                          I
```

Ausgabe eines Zeichens aus dem CPU-Register E auf den sequentiellen Kanal.

```
I                                          I
I  Funktion 5: Ausgabe auf Listgeraet      I
I                                          I
I   Aufrufparameter:                       I
I         Register  C:   05H               I
I         Register  E:   Zeichen           I
I                                          I
```

Das Zeichen aus dem CPU-Register E wird auf dem Listgeraet ausgegeben.

3.4. Arbeit mit Diskettendateien

Bei fast allen diesen Funktionen wird ein Parameterblock benoetigt, der alle Informationen zur Diskettenarbeit enthaelt. Dieser Block wird Dateisteuerblock (FCB) genannt. Meist ist seine Adresse vor dem BDOS-Aufruf im Registerpaar DE einzustellen.
Die gesamte Diskette ist in Saetze mit je 128 Byte aufgeteilt. Bei Datenmanipulation mit der Diskette kann immer nur auf Saetze zugegriffen werden. BDOS unterstuetzt die Organisation von Dateien, die aus einer Menge dieser Saetze besteht. Alle Parameter der Dateien und fuer den Zugriff befinden sich im FCB.

3.4.1. Aufbau des FCB

Byte	Kurzbezeichnung	Bedeutung
+0	dr	Laufwerkscode 0...aktuelles LW (siehe auch Funktion 14) 1...LW A 2...LW B 3...LW C ..... 16 ...LW P
+1...8	f1...f8	Dateiname Die siebenten Bit dieser Bytes sind fuer den Dateinamen nicht signifikant. Hier koennen Nutzerflags f...1 - f...8 angelegt werden.
+9...11	t1...t3	Dateityp Die Bits 7 von t1 und t2 haben folgende Bedeutung: Bit. 7 von t1 = 1 ---> die Datei ist eine Nur-Lese-Datei (R/O). Bit 7 von t2 = 1 ---> die Datei ist eine SYS-Datei. (siehe dazu auch "Anleitung fuer den Bediener" Pkt.3.7) ???
12	ex	aktuelle Bereichs(Extent-)-Nummer

13,14		fuer SCPX reserviert
		Fuellstand des aktuellen Extent (Wertebereich: 0-128) in Anzahl Saetzen
16...31		fuer SCPX reserviert
32	cr	aktuelle Satznummer des Extent fuer den naechsten sequentiellen Zugriff
33—35	r0,r1,r2	aktuelle Satznummer der Datei (nur fuer Direktzugriff erforderlich!) r0- L-Teil) r1- H-Teil) der Satznummer (Wertebereich: 0...65535) r2- fuer Ueberlauf

3.4.2. Verwaltung der Verzeichniseintragungen

Zur Verwaltung der Dateien befindet sich auf der Diskette ein Bereich mit einem Verzeichnis, in das die aktuellen Zeiger der Dateien eingetragen sind. Beim Zugriff auf Dateien werden, von diesen Eintragungen ausgehend, die Daten manipuliert.
Zur Verwaltung dieser Eintragungen sind die folgenden BDOS-Funktionen installiert:

Funktion 22: Anlegen Datei
Aufrufparameter:
Register C: 16H
Register DE: FCB-Adresse
Rueckgabewert:
Register A: Verzeichnis-code

Ist eine neue Datei anzulegen, so ist diese Funktion zu verwenden. Die BDOS-Funktion #15 (OPEN) ist nur fuer bereits existierende Dateien wirksam! Im Registerpaar DE ist die FCB-Adresse vorzugeben.
Da bei dieser Funktion keine Kontrolle auf eine bereits unter gleicher Bezeichnung bestehende Datei erfolgt, sollte sicherheitshalber vorher die BDOS-Funktion #19 (Loeschen einer Datei) ausgefuehrt werden.
Die Datei wurde erfolgreich in dem Verzeichnis angelegt, wenn im Register A als Rueckmeldung der Wert 00H bis 03H steht. Eine 0 im CPU-Register A sagt aus, dass in dem Verzeichnis keine Eintragung mehr frei ist.
Nach der Funktion #22 ist die Funktion #15 (OPEN) nicht erforderlich!

Funktion 15: Eroeffnen einer Datei
Aufrufparameter:
Register C: 0FH
Register DE: FCB-Adresse
Rueckgabewert:
Register A: Verzeichnis-code

Im Registerpaar DE steht die Adresse des FCB. Mit dieser Funktion wird die im FCB benannte Datei (Bytes f1...f8, t1...t3), die auf dem durch das Byte 0 des FCB ausgewaehlten Laufwerks existieren muss, aktiviert. Dabei werden bestimmte Werte aus der Verzeichnis-Eintragung in den FCB uebernommen, so dass anschliessend die Funktionen fuer das Lesen und Schreiben ausgefuehrt werden koennen.

Wird die Datei gefunden, so wird im Register A der Wert 00H, 01H, 02H oder 03H gesetzt.Enthaelt das Register A den Wert FFH, wurde der Suchbegriff des FCB nicht gefunden.
Im Namen der Datei im FCB koennen auch Fragezeichen (Codierung 3FH) enthalten sein. Fuer sie kann jedes andere beliebige Zeichen im Namen der Datei in dem Verzeichnis an der gleichen Position stehen. Es wird die erste Datei in dem Verzeichnis aktiviert, die mit dem Zeichen in den anderen Positionen ueber-einstimmt.
Soll nach dem Eroeffnen der Datei sequentiell auf den ersten Satz zugegriffen werden, so ist das Byte "cr" im FCB auf 00H zu setzen.

Funktion 16: Schliessen einer Datei
Aufrufparameter:
Register C: 10H
Register DE: FCB-Adresse
Rueckgabewert:
Register A: Verzeichnis-code

Im Registerpaar DE ist die Adresse des FCB anzugeben. Mit dieser Funktion werden die im FCB gespeicherten Informationen in die entsprechende Verzeichnis-Eintragung uebernommen. Die Datei muss aber vorher durch die Funktion 15 oder 22 aktiviert worden sein. Wurde von der Datei nur gelesen, ist kein Schliessen erforderlich, aber nach Schreiboperationen ist die Funktion zur Aktualisierung der Verzeichniseintragung auf der Diskette unbedingt erforderlich.
Steht nach dem Funktionsaufruf im CPU-Register A ein Wert zwischen 00H und 03H, so war die Funktion erfolgreich. Wurde der Name der Datei nicht gefunden, so enthaelt CPU-Register A den Wert FFH.

Funktion 17: Suchen nach erster Eintragung	
Aufrufparameter:	
Register C:	11H
Register DE:	FCB-Adresse
Rueckgabewert:	
Register A:	Verzeichnis-code

Mit dieser Funktion werden dem Benutzer (vor allem dem Systemprogrammierer) die Verzeichnis-Eintragungen zugaenglich gemacht. Das Verzeichnis wird von Anfang an nach einer Eintragung entsprechend der FCB-Eintragungen dr, f1...f8, t1...t3, ex durchsucht.
Das Registerpaar DE enthaelt wieder die FCB-Adresse. Fragezeichen im Namen lassen an diesen Positionen jeden Buchstaben zu. Ein Fragezeichen im Byte 0 des FCB (dr) ermoeglicht das Durchsuchen nach allen Eintragungen, auch aller USER-Bereiche, im aktuellen Laufwerk.
Ein erfolgreiches Suchen wird im Register A durch die Werte 00H, 01H, 02H oder 03H angezeigt. Im anderen Fall enthaelt das Register den Wert FFH.
Die Eintragung, die entsprechend der FCB-Vorgaben ermittelt wurde und 32 Byte lang ist, wird in einem 128 Byte-Puffer eingetragen. Dieser beginnt entweder bei 0080H oder bei einer mit der Funktion 26 vorgegebenen Adresse.
Je nach dem Wert von A ist die Eintragung ab dem ersten Byte (A=00H), 33.Byte (A=01H), 65. Byte (A=02H) oder 97. Byte (A=03H) innerhalb des Puffers abgelegt.

```
I                                          I
I  Funktion 18: Suchen nach der            I
I               naechsten Eintragung       I
I                                          I
I   Aufrufparameter:                       I
I         Register  C:   12H               I
I                                          I
I   Rueckgabewert:                         I
I         Register  A:   Verzeichnis-      I
I                        code              I
I                                          I
```

Diese Funktion wirkt aehnlich der Funktion #17. Das Durchsuchen des Verzeichnisses erfolgt aber ab der aktuellen Position. Diese Funktion ist im Anschluss an die Funktion #17 zu programmieren, wenn nach weiteren Dateien (im Namen des FCB sind Fragezeichen enthalten) oder weiteren Eintragungen der gleichen Datei (grosse Dateien koennen mehrere Eintragungen belegen) gesucht wird.

```
I                                          I
I  Funktion 19: Loeschen einer Datei       I
I                                          I
I   Aufrufparameter:                       I
I         Register  C:   13H               I
I         Register DE:   FCB-Adresse       I
I                                          I
I   Rueckgabewert:                         I
I         Register  A:   Verzeichnis-      I
I                        code              I
I                                          I
```

Im Registerpaar DE ist die FCB-Adresse vorzugeben. Mit dieser Funktion wird die Datei auf der Diskette geloescht. Enthaelt der Dateiname oder Dateityp im FCB ein oder mehrere Fragezeichen, so werden alle Dateien geloescht, die in den anderen Positionen gleiche Zeichen enthalten.
Steht als Rueckmeldung im Register A ein ØFFH, so wurde keine Datei auf der Diskette gefunden, die der Dateibezeichnung im FCB entspricht.
Erfolgreiches Loeschen einer Datei wird mit den Werten ØØH, Ø1H, Ø2H oder Ø3H im Register A zurueckgemeldet.

Funktion 23: Umbenennen einer Datei
Aufrufparameter:
Register C: 17H
Register DE: FCB-Adresse
Rueckgabewert:
Register A: Verzeichnis-code

Mit dieser Funktion kann auf der Diskette eine Datei umbenannt werden.
Das Registerpaar DE adressiert einen FCB, der etwas anders aufgebaut ist als bei den anderen Funktionen:

Die Bytes 0-15 enthalten im ueblichen Format die "alte" Dateibezeichnung einschliesslich Laufwerksangabe, die Bytes 16-32 die "neue" Dateibezeichnung. Hier ist die Laufwerksangabe bedeutungslos. Im CPU-Register A steht nach erfolgreicher Ausfuehrung ein Wert zwischen 00H und 03H. Wenn die umzubennende Datei nicht gefunden wurde, so steht im CPU-Register der Wert FFH.

3.4.3. Adressierung und Datentransport

Fuer den Datentransport sind folgende Parametereinstellungen notwendig:

- Adressierung des Speicherbereiches fuer den 128-Byte-Puffer (DMA-Adressierung)
- Adressierung des zu transportierenden Satzes
- Richtung des Transportes (Lesen/Schreiben)

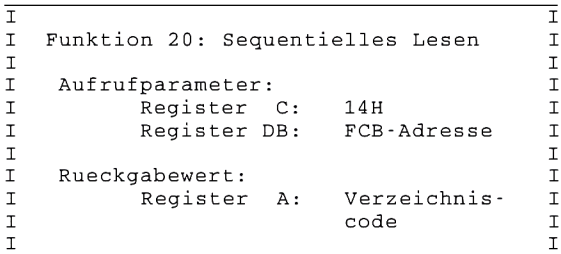
Die Adressierung des zu transportierenden Satzes wird entweder dadurch vorgenommen, dass eine beliebige Satznummer vorgewaehlt und in den erweiterten FCB eingetragen wird (wahlfreier Zugriff), oder die Satznummer von BDOS selbst verwaltet wird (sequentieller Zugriff). Im letzten Fall wird der Zeiger im FCB, der die aktuelle Satznummer beinhaltet, bei sequentiellem Zugriff auf den nachfolgenden Satz der Datei gestellt.

Die DMA-Adresse wird durch die Transportoperationen nicht veraendert. Bei direktem Zugriff wird der Satzzeiger nicht geaendert, aber der sequentielle Zeiger verweist auf den Zugriffspunkt. Damit ist ein Uebergang von direktem zu sequentiellen Zugriff moeglich. Bei Zugriffswechsel in umgekehrter Richtung existiert eine zusaetzliche Transformationsfunktion. Durch diese wird der Satzzeiger im FCB fuer direkten Zugriff, entsprechend dem Stand des sequentiellen Zeigers, aktualisiert. Aufgrund der dynamischen Verwaltung der Diskettenbelegung bei SCPX wird auch die Funktion des Schreibens von Saetzen mit Blockinitialisierung angeboten.
Es stehen folgende BDOS-Funktionen zur Verfuegung:

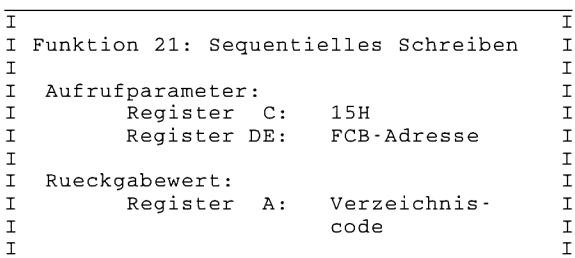
Funktion 26: Einstellen DMA-Adresse
Aufrufparameter:
Register C: 1AH
Register DE: DMA-Adresse

Fuer den Datentransport von und zur Diskette ist ein Puffer mit 128 Byte Kapazitaet erforderlich. Nach Kalt-, Warmstart und Reset des Diskettensystem (BDOS-Funktion #13) hat dieser Puffer die Adresse 0080H.

Durch diese Funktion kann eine andere Adresse fuer diesen Puffer eingestellt werden. Sie ist im Registerpaar DE vorzugeben. Diese DMA-Adresse bleibt wirksam bis zur naechsten BDOS-Funktion #26, BDOS-Funktion #13 oder bis zum naechsten Kalt- oder Warmstart.



Im Registerpaar DE ist die FCB-Adresse vorzugeben. Es werden mit dieser Funktion 128 Byte (ein Satz) von dieser Datei in den Speicher eingelesen. Die Speicheranfangsadresse (DMA-Adresse) ist entweder 0080H oder sie wird durch die BDOS-Funktion #26 vorgegeben. Die Angabe "cr" im FCB gibt den naechsten zu lesenden Satz an (cr=00H bedeutet also, es wird der 1. Satz gelesen). Nach dem Lesen wird "cr" automatisch um 1 erhoeht. Wird die Extentgrenze erreicht, so wird auch automatisch auf den folgenden Extent umgeschaltet und "cr" auf 00H gesetzt. Wird im Register A ein Wert ungleich 00H gemeldet, so ist das Dateiende erreicht. Bei erfolgreichem Lesevorgang steht im CPU-Register A der Wert 00H.



Im Registerpaar DE ist die FCB-Adresse vorzugeben. Von der aktuellen DMA-Adresse wird ein Satz in die durch FCB vorgegebene Datei uebertragen, wobei "cr" die Position des Satzes in der Datei festlegt. Nach dem Schreiben wird "cr" automatisch um 1 erhoeht. Das Umschalten auf einen naechsten Extent erfolgt ebenfalls automatisch, "cr" wird dabei auf 00H gesetzt.
Ein erfolgreicher Schreibversuch wird mit 00H im CPU-Register A zurueckgemeldet. Ist die Diskette gefuellt und der Satz kann nicht mehr aufgezeichnet werden, so steht im CPU-Register A ein Wert ungleich 00H.

Beachte:
Nach dem Schreiben muss die CLOSE-Funktion folgen, damit die Informationen des Verzeichnisses auf der Diskette aktualisiert werden.

Lese - und Schreibfehler fuehren zu der Konsolmeldung:

SCPX ERR ON d: BAD SECTOR

(d gibt das logische Laufwerk an: A, B, ...P).

Mit der Abschlusstaste wird die Fehlermeldung ignoriert und das Programm fortgesetzt. Dabei steht im CPU-Register A ein Wert ungleich 00H!
Es besteht ausserdem die Moeglichkeit, mit ^C das Programm durch Sprung zum Warmstart zu beenden.

Funktion 33: Lesen mit direktem Zugriff		
Aufrufparameter:		
	Register C:	21H
	Register DE:	FCB-Adresse
Rueckgabewert:		
	Register A:	Rueckkehr-code

Fuer den direkten Zugriff sind die Bytes r0, r1 und r2 im FCB erforderlich, dessen Adresse im Registerpaar DE stehen muss. In r0 ist der L-Teil und in r1 der H-Teil der Adresse des Satzes einzutragen, der gelesen werden soll. Das Byte r2 ist auf 00H zu setzen.
Bei den vorher erforderlichen BDOS-Funktionen #15 bzw. #22 (OPEN, Anlegen Datei) ist zu garantieren, dass im adressierten FCB das Byte 12 (ex) auf 00H gesetzt ist.
Der Satz wird im durch die DMA-Adresse vorgegebenen Puffer eingetragen.

Im Gegensatz zum sequentiellen Lesen wird die Satzadresse nach dem Zugriff nicht um 1 erhoeht, so dass eine sofort folgende BDOS-Funktion #33 den gleichen Satz lesen wuerde.
Nach dem Lesen werden durch diese Funktion ausserdem die aktuelle Satznummer "cr" und die Extentnummer "ex" gesetzt. Damit ist ein anschliessendes sequentielles Arbeiten moeglich.
Es ist zu beachten, dass der zuletzt gelesene Satz eingestellt ist, d.h. er wird dann nochmals gelesen bzw. ueberschrieben.
Bei erfolgreichem Lesen wird im Register A eine 00H gemeldet.
Da nicht alle Satzadressen beim direkten Zugriff zwischen niedrigstem und hoechstem Wert existieren muessen (siehe BDOS—Funktion #34), kann es zu folgenden Fehlermeldungen im CPU-Register A kommen:

01H Es sollte ein Satz mit unbeschriebenen Daten gelesen werden.

03H Vor diesem Funktionsaufruf wurde die Funktion Schreiben mit direktem Zugriff ausgefuehrt. Das jetzt eventuell erforderliche Aktualisieren der Verzeichnis-Eintragung, durch das vorherige Schreiben hervorgerufen, konnte nicht ausgefuehrt werden.

04H Es sollte ein Satz von einem unbeschriebenen Extent gelesen werden.

06H Mit der eingestellten Satzadresse wuerde das Diskettenende ueberschritten werden.

Funktion 34: Schreiben mit direktem Zugriff		
Aufrufparameter:		
	Register C:	22H
	Register DE:	FCB-Adresse
Rueckgabewert:		
	Register A:	Rueckkehr-code

Wie beim Lesen mit direktem Zugriff geben das Registerpaar DE die FCB-Adresse und dort die Bytes r0 und r1 die Adresse des Satzes an. An diese Adresse wird der Satz aus dem durch die DMA-Adresse vorgegebenen Puffer geschrieben und falls ein neuer Extent dafuer erforderlich ist, dieser Extent in das Verzeichnis eingetragen.

Die Satzadresse wird wie beim Lesen nicht erhoeht.
Es wird aber die aktuelle Satznummer "cr" und die Extentnummer "ex" im FCB aktualisiert, so dass ein anschliessendes sequentielles Arbeiten (wie beim Lesen mit direktem Zugriff beschrieben) moeglich ist.

Beachte:
Sobald ein Satz in einen SCP-Diskettenblock der Datei geschrieben wurde, gelten alle anderen Saetze in diesem Block als geschriebene Daten. Das Lesen eines dieser Bloecke bringt also keinen Fehlercode 01H.
Die Fehlercodes sind die gleichen wie beim Lesen mit direktem Zugriff (ausser 01H und 04H), dazu kommt der Fehlercode 05H. Diese Fehlermeldung signalisiert, dass in dem Verzeichnis fuer die erforderliche neue Eintragung kein Platz mehr frei ist.

Funktion 40:	Schreiben mit direktem Zugriff und Blockinitialisierung	
Aufrufparameter:		
	Register C:	28H
	Register DE:	FCB-Adresse
Rueckgabewert:		
	Register A:	Rueckkehrcode

Diese Funktion entspricht der BDOS-Funktion #34 mit der Erweiterung, dass ein neuangelegter Block automatisch initialisiert wird. Alle Saetze des Blockes werden mit dem Wert 0 vorbelegt

```
I                                                I
I   Funktion. 36: Bereitstellen aktuelle     I
I                 Satzposition                   I
I                                                I
I   Aufrufparameter:                             I
I         Register  C:   24H                     I
I         Register DE:   FCB-Adresse             I
I                                                I
I   Rueckgabewert:                               I
I         Satzadressen des letzten               I
I         sequentiellen Zugriffs                 I
I                                                I
```

Das Registerpaar DE adressiert den FCB, in dessen Bytes r0 und r1 durch diese Funktion die Satzadressen des letzten sequentiellen Zugriffes abgelegt werden. Damit kann z.B. auf einfache Weise vom sequentiellen zum direkten Zugriff uebergegangen werden.

3.4.4. System- und Hilfsfunktionen

Mit diesen Funktionen werden die Initialisierungen des Betriebssystems ausgefuehrt bzw. die Abfrage und Beeinflussung von Betriebssystemzustaenden ermoeglicht.
Zu unterteilen ist in
- Systemfunktionen
- Laufwerk-Funktionen
- Dateiverwaltungsfunktionen.

3.4.4.1. Systemfunktionen

```
I                                        I
I Funktion 0: System - Neustart          I
I                                        I
I Aufrufparameter:                       I
I Register C: 00H                        I
I                                        I
```

Mit dieser Funktion erfolgt die Rueckkehr in den CCP-Status durch einen Warm-Start. Die Wirkung entspricht der Funktion Warmstart des BIOS-Sprungverteilers. Die BDOS-Funktion verzweigt zur absoluten Speicheradresse 0000H. Dort ist ein Sprung in den BIOS-Warmstart eingetragen.
Beim Warmstart wird die DMA-Adresse 80H eingestellt und anschliessend die Systemsteuerung an das CCP uebergeben.

Funktion 12: Abfrage Versionsnummer
Aufrufparameter:
Register C: 0CH
Rueckgabewert:
Register HL: Versionsnummer

Mit dieser Funktion kann das Nutzerprogramm die Leistungsfaehigkeit des SCPX abfragen. Als Antwort erhaelt das aufrufende Programm im CPU-Doppelregister HL den Wert 0022. Das symbolisiert eine Leistungsfaehigkeit, die der vom CP/M 2.2 entspricht.

Funktion 32: Einstellen/Abfragen Benutzercode	
Aufrufparameter:	
Register C:	20H
Register E:	0FFH (Abfragen) oder Benutzercode (Einstellen)
Rueckgabewert:	
Register A:	aktueller Benutzercode oder kein Wert

Wird im CPU-Register E ein FFH eingetragen, so wird nach dem Funktionsaufruf im CPU-Register A der aktuelle Benutzercode zurueckgemeldet.
Ist der Wert im Register ungleich FFH, so werden durch die Funktion 32 die unteren 5 Bit dieses Wertes als neuer Benutzercode eingestellt (00H...1FH).

Beachte:
Mit dem residenten Kommando USER lassen sich nur die Benutzerbereiche 0-15 (00H...0FH) einstellen, so dass die anderen Kommandos nur in diesen Benutzerbereichen wirksam werden koennen. Es sollte deshalb generell nur mit dem Benutzercode 00H...0FH gearbeitet werden.

```
I                                                    I
I   Funktion 31: Bereitstellen Adresse der           I
I                 Laufwerk - Parameter               I
I                                                    I
I   Aufrufparameter:                                 I
I         Register  C:   1FH                         I
I                                                    I
I   Rueckgabewert:                                   I
I         Register HL:   DPB - Adresse               I
I                                                    I
```

Nach dem Funktionsaufruf steht im CPU-Registerpaar HL die Adresse der Parametertabelle des aktuellen Laufwerkes. Diese Tabelle ist im BIOS-Teil angelegt und enthaelt Organisationsparameter des entsprechenden Laufwerkes, die auch von BDOS herangezogen werden. In diesem Parameterfeld kann der Systemprogrammierer spezielle Informationen auswerten oder Manipulationen der Parameter durchfuehren.

```
I                                                    I
I   Funktion 27: Bereitstellen Adresse des           I
I                 Allocation - Vektors               I
I                                                    I
I   Aufrufparameter:                                 I
I         Register  C:   1BH                         I
I                                                    I
I   Rueckgabewert:                                   I
I         Register HL:   Adresse des                 I
I                        Allocation-Vektors          I
I                                                    I
```

Fuer jedes logische Laufwerk, auf das einmal zugegriffen wurde, gibt es im Speicher eine Tabelle, die die Belegung der Diskette (belegte Bloecke - freie Bloecke) widerspiegelt.
Mit dieser Funktion wird im Registerpaar HL die Adresse dieser Tabelle (Allocation-Vektor) fuer das aktuelle Laufwerk bereitgestellt.
Diese Funktion ist normalerweise nur fuer den Systemprogrammierer von Bedeutung.

3.4.4.2. Laufwerksfunktionen

Im SCPX sind 16 logische Laufwerke A bis P definiert. Die BDOS-Funktionen arbeiten mit diesen 16 Laufwerken. Dabei ist zu gewaehrleisten, dass im BIOS alle die Laufwerke installiert sind, die vom BDOS-Aufrufer angefordert werden.
Im BDOS existiert ein ausgezeichnetes Laufwerk, das Laufwerk A, und ein aktuelles Laufwerk, das selektiert wird und bis zur Selektion eines anderen Laufwerkes selektiert bleibt. Dieses Laufwerk hat eine bevorzugte Stellung; die Parameteruebergabe fuer andere BDOS-Funktionen bezieht sich auf diese Selektion. Jedes Laufwerk, auf das nach einer Initialisierung zugegriffen wird, erhaelt den "on-line-Zustand", alle anderen bleiben im "off-line-Zustand". Jedes Laufwerk kann durch Setzen des entsprechenden Schreibschutzes in den "read-only-Zustand" versetzt werden.

Funktion 13: Ruecksetzen Diskettensystem
Aufrufparameter:
Register C: 0DH

Um in einem Programm Disketten ohne Warmstart (der zum Programmabbruch fuehrt) wechseln zu koennen, ist diese Funktion erforderlich. Sonst fuehrt ein Diskettenwechsel ohne Warmstart oder ohne anschliessende BDOS-Funktion #13 zu R/O-Fehler. Mit dieser Funktion wird das LW A das aktuelle Laufwerk, alle Laufwerke erhalten den R/W-Status (siehe BDOS-Funktion #28), die aktuelle DMA-Adresse wird auf 0080H eingestellt (siehe BDOS-Funktion #26) und alle Laufwerke ausser LW A erhalten das "off-line-Attribut".

```
I                                                  I
I   Funktion 37: Laufwerke zuruecksetzen           I
I                                                  I
I   Aufrufparameter:                               I
I         Register  C:   25H                       I
I         Register DE:   Reset-Marke               I
I                                                  I
I   Rueckgabewert:                                 I
I         Register  A:   00H                       I
I                                                  I
```

Mit dieser Funktion koennen Laufwerke einzeln zurueckgesetzt werden (im Gegensatz zu BDOS-Funktion #13, die alle logischen Laufwerke zuruecksetzt). Damit koennen einzelne Laufwerke zurueckgesetzt werden, um entweder Disketten in den rueckgesetzten Laufwerken wechseln zu koennen oder den Schreibschutz fuer diese Laufwerke zu loeschen. Im CPU-Doppelregister DE wird die Reset-Marke eingetragen. Jedem der 16 Bit ist ein logisches Laufwerk A,..., P in folgender Art
Bit 0, E : LW A
Bit 1, E : LW B
.....
Bit 7, D : LW P
zugeordnet. Die 1-Belegung kennzeichnet die zurueckzusetzenden Laufwerke.

```
I                                              I
I   Funktion 14: Laufwerk selektieren          I
I                                              I
I   Aufrufparameter:                           I
I         Register  C:   0EH                   I
I         Register  E:   Laufwerk              I
I                                              I
```

Der Inhalt vom Register E gibt an, welches logische Laufwerk fuer die folgenden Dateioperationen das aktuelle LW ist.
Dabei gilt die folgende Zuordnung:
E=00 : LW A
E=01 : LW B
.....
E=15 : LW P
Ist z.B. mit dieser Funktion das LW C selektiert, so bedeutet der Wert 00 im ersten Byte des FCB, dass mit diesem FCB bei Dateizugriff die Diskette im Laufwerk C angesprochen wird. Das ausgewaehlte Laufwerk bleibt jetzt aktiviert (im "on-line-Zustand") bis zum naechsten Ruecksetzen Diskettensystem oder Warmstart. Bei einem Diskettenwechsel waehrend des "on-line-Zustandes" wuerde das Laufwerk in den R/O-Status gehen.

```
I                                                    I
I   Funktion 25: Abfrage nach aktuellem              I
I                Laufwerk                            I
I                                                    I
I   Aufrufparameter:                                 I
I         Register  C:   19H                         I
I                                                    I
I   Rueckgabewert:                                   I
I         Register A:    aktuelles Laufwerk          I
I                                                    I
```

Diese BDOS-Funktion meldet im CPU-Register A das aktuelle logische Laufwerk. Es gilt die Zuordnung:

A=00 : LW A
A=1 : LW B usw.

```
I                                                    I
I   Funktion 29: Abfrage nach Laufwerk mit           I
I                Schreibschutz                       I
I                                                    I
I   Aufrufparameter:                                 I
I         Register  C:   1DH                         I
I                                                    I
I   Rueckgabewert:                                   I
I         Register HL:   Schreibschutzwert           I
I                                                    I
```

In den CPU-Registern H und L wird wie bei BDOS-Funktion #24 ein Code abgelegt, aus dem zu entnehmen ist, welches logische Laufwerk den R/O(read only)-Status hat. Das diesen Laufwerken zugeordnete Bit ist auf "1" gesetzt.
Der R/O-Status kann durch die BDOS-Funktion #28 und durch einen Diskettenwechsel (ohne folgende BDOS-Funktion #13) gesetzt sein.

```
I                                                    I
I   Funktion 24: Abfrage nach Laufwerk im        I
I                 "on-line-Zustand"                  I
I                                                    I
I   Aufrufparameter:                                 I
I         Register  C:   18H                         I
I                                                    I
I   Rueckgabewert:                                   I
I         Register HL:   Zustandswert                I
I                                                    I
```

Mit dieser BDOS-Funktion laesst sich feststellen, mit welchen Laufwerken seit der letzten Systeminitialisierung (Kaltstart, Warmstart, Reset des Disk-Systems) gearbeitet wurde.In den CPU-Registern H und L wird dieses zur Verfuegung gestellt. In diesem 16-Bit-Vektor wird jedes Bit einem LW folgendermassen zugeordnet:

```
Bit 0,L     : LW A
BIT 1,L     : LW B
.................
Bit 7,L     : LW H
Bit 0,H     : LW I
.................
Bit 7,H     : LW P
```

Der Bitwert 0 stellt den "off-line-Zustand",
der Bitwert 1 stellt den "on-line-Zustand" dar.

3.4.4.3. Dateienverwaltung

```
I                                                    I
I   Funktion 30: Setzen Datei - Attribute        I
I                                                    I
I   Aufrufparameter:                                 I
I         Register  C:   1EH                         I
I         Register DE:   FCB-Adresse                 I
I                                                    I
I   Rueckgabewert:                                   I
I         Register  A:   Verzeichniscode             I
I                                                    I
```

Mit dieser Funktion koennen Dateien mit den Attributen R/O (Nur-Lese-Datei) oder R/W (Lese-Schreib-Datei) und DIR (Datei wird beim residenten Kommando DIR angezeigt) oder SYS (keine Anzeige bei DIR) versehen werden (siehe auch "Anleitung fuer den Bediener", Pkt. 3.7.).

Das CPU-Registerpaar DE adressiert einen FCB. Ist das Bit 7 von t1 gesetzt, so wird die Datei mit dieser Funktion zu einer R/O-Datei. Ist das Bit nicht gesetzt, wird die Datei zu einer R/W-Datei.
Ein gesetztes Bit 7 von t2 versetzt die Datei in den SYS-Datei-Status. Im anderen Fall erhaelt die Datei das DIR-Attribut.

```
I                                                I
I   Funktion 35: Bereitstellen naechste          I
I                freie Satzposition              I
I                                                I
I   Aufrufparameter:                             I
I       Register  C:     23H                     I
I       Register DE:     FCB-Adresse             I
I                                                I
I   Rueckgabewert:                               I
I         Adresse nach letztem Satz              I
I                                                I
```

Mit dem CPU-Registerpaar DB ist ein FCB zu adressieren. Nach Ausfuehrung der Funktion enthalten die Bytes r0 und r1 die Adresse nach dem letzten Satz der Datei. Damit ist es einfach moeglich, eine Datei mit Direktzugriff weiterzuschreiben.

3.5 Zusammenfassung der BDOS-Funktionen

3.5.1. Uebersicht nach Funktionsgruppen geordnet

Bezeichnung	Nr.	Eingang	Ausgang
= Zeichenorientierte Ein- und Ausgabe			
- Zuordnung der Kanaele			
IOBYTE abfragen	7	- - -	A: IOBYTE
IOBYTE belegen	8	E: IOBYTE	- - -
- Daten ueber Konsole			
Konsoleneingabe + Echo	1	— - -	A: Zeichen (*1)
Konsolenausgabe	2	E: Zeichen (mit ^S, ^P)	- — -
Konsolenstatus	11	- - —	A: Konsolen-Status (*2)
direkte Konsolen E/A	6	E = FF: Status	A: Konsolen-Status (*2)
		E = /FF: Ausgabe Zeichen	- - -
Zeichenkette ausgeben	9	DE -> Kette EKZ=¤, mit ^S,^P	- - -
Eingabe Konsolpuffer	10	DE -> Puffer (*3)	- - -
- Daten ueber sequentiellen Kanal und Listgeraet			
Sequentielle Eingabe	3	- - -	A: Zeichen
Sequentielle Ausgabe	4	E: Zeichen	- - -
List — Ausgabe	5	E: Zeichen	- - -

Bezeichnung	Nr.	Eingang	Ausgang
= Arbeit mit Diskettendateien			
- Verwaltung des Dateienverzeichnis			
Datei erzeugen	22	DE -> FCB	A: DC(*7, =FF: Verz. voll)

Datei eroeffnen	15	DE -> FCB	A: DC(*7, =FF:keine Datei)
Datei schliessen	16	DE -> FCB	A: DC(*7, =FF:keine Datei)
erste Eintragung suchen	17	DE -> FCB	A: DC(*7, =FF:keine Datei)
folgende Eintragung suchen	18		A: DC(*7, =FF:keine Datei)
Dateien loeschen	19	DE -> FCB	A: DC(*7, =FF:keine Datei)
Datei umbenennen	23	DE -> FCB (*8)	A: DC(*7, =FF:keine Datei)

- Adressierung und Datentransport

Datenpuffer adressieren	26	DE -> Puffer	- - -
naechsten Satz lesen	20	DE -> FCB	A: EC(*5, =/00:EOF)
naechsten Satz schreiben	21	DE -> FCB	A: EC(*5, =/00:Disk. voll)
direkt adressierten Satz lesen	33	DE -> FCB	A: RC(*6)
direkt adressierten Satz schreiben	34	DE -> FCB	A: RC(*6)
direkt adressierten Satz schreiben und Block initialisieren	40	DE -> FCB	A: RC(*6)
Berechnung der aktuellen Satzadresse	36	DE -> FCB	Eintrag in FCB + 33,34,35: r0-2=fkt(ex,cr)

= System- und Hilfsfunktionen

- Systemfunktionen

Warmstart	0	- - -	- - -
Version ermitteln	12	- - -	HL: Version

Bezeichnung	Nr.	Eingang	Ausgang
Nutzernummer abfragen/setzen	32	E= FF: Abfrage E= /FF: Setzen	A:Nutzer# (0-15)
Diskettenparametertabelle (DPB) ermitteln	31	- - -	HL -> DPB des selekt. LW
Belegungstabelle (ALLOC) ermitteln	27	- - -	HL -> ALLOC

- Laufwerkfunktionen

Diskettensystem zuruecksetzen	13	- - -	A: Batchmode - Flag (*9)
ausgewaehlte Laufwerke zuruecksetzen	37	DE: LW-Vektor	A: 00
Auswahl Bezugs-LW	14	E: LW# (0...F)	- - -
Abfrage Bezugs-LW	25	- - -	A: LW# (0 - 15)
Abfrage geschuetzte LW	29	- - -	HL: LW - Vektor
Schutz des Bezugs-LW	28	— — —	— — —
Abfrage angeschlossener LW	24	- - -	HL: LW - Vektor

- Dateienverwaltung

Dateimerkmale setzen	30	DE -> FCB	A: DC(*7, =FF:keine Datei)
Dateigroesse berechnen	35	DE -> FCB	- - -

3.5.2. Uebersicht nach Auftragsnummern geordnet

Nr.	Bezeichnung	Eingang	Ausgang
0	Warmstart	- - -	- - -
1	Konsoleneingabe + Echo	- - -	A: Zeichen (*1)
2	Konsolenausgabe	E: Zeichen (mit ^S, ^P)	- - -
3	Sequentielle Eingabe	- - -	A: Zeichen
4	Sequentielle Ausgabe	E: Zeichen	— — —
5	List - Ausgabe	E: Zeichen	- - -
6	direkte Konsolen E/A	E = FF: Status	A: Konsolenstatus (*2)
		E = /FF: Ausgabe Zeichen	- - -
7	IOBYTE abfragen	- - -	A: IOBYTE
8	IOBYTE belegen	E: IOBYTE	- - -
9	Zeichenkette ausgeben	DE -> Kette EKZ=¤, mit ^S,^P	- - -
10	Eingabe Konsolpuffer	DE -> Puffer (*3)	- - -
11	Konsolenstatus	— — —	A: Konsolenstatus (*2)
12	Version ermitteln	- - -	HL: Version
13	Diskettensystem zuruecksetzen	— — —	A: Batchmode - Flag (*9)
14	Auswahl Bezugs-LW	E: LW# (0...F)	- - -

15	Datei eroeffnen	DE -> FCB	A: DC(*7, =FF:keine Datei)
16	Datei schliessen	DE -> FCB	A: DC(*7, =FF:keine Datei)
17	erste Eintragung suchen	DE -> FCB	A: DC(*7, =FF:keine Datei)
18	folgende Eintragung suchen	- - -	A: DC(*7, =FF:keine Datei)
19	Dateien loeschen	DE -> FCB	A: DC(*7, =FF:keine Datei)
20	naechsten Satz lesen	DE -> FCB	A: EC(*5, =/00: EOF)
21	naechsten Satz schreiben	DE -> FCB	A: EC(*5, =/00:Disk. voll)
22	Datei erzeugen	DE -> FCB	A: DC(*7, =FF: Verz. voll)
23	Datei umbenennen	DE -> FCB (*8)	A: DC(*7, =FF:keine Datei)
24	Abfrage ange- schlossener LW	- - -	HL: LW - Vektor
25	Abfrage Bezugs-LW	- - -	A: LW# (0 - 15)
26	Datenpuffer adressieren	DE -> Puffer	- - -
27	Belegungstabelle (ALLOC) ermitteln	- - -	HL -> ALLOC
28	Schutz des Bezugs-LW	- - -	- - -
29	Abfrage geschuetzte LW	- - -	HL: LW - Vektor
30	Dateimerkmale setzen	DE -> FCB	A: DC(*7, =FF:keine Datei)
31	Diskettenparameterta belle (DPB) ermitteln	- - -	HL -> DPB des selekt. LW
32	Nutzernummer abfragen/setzen	E= FF: Abfrage E= /FF: Setzen	A:Nutzer# (0-15)
33	direkt adressierten Satz lesen	DE -> FCB	A: RC(*6)
34	direkt adressierten Satz schreiben	DE -> FCB	A: RC(*6)
35	Dateigroesse berechnen	DE -> FCB	- - -
36	Berechnung der aktuellen Satzadresse	DE -> FCB	Eintrag in FCB + 33,34,35: r0-2=fkt(ex,cr)
37	ausgewaehlte Laufwerke zuruecksetzen	DE: LW-Vektor	A: 00
40	direkt adressierten Satz schreiben und Block initialisieren	DE -> FCB	A: RC(*6)

3.5.3. Erlauterungen und Kommentare

Erlaeuterungen:

- Die Zeichen "->" sind zu interpretieren als "Zeiger auf"
- Die Zeichen "=/" sind zu interpretieren als "ungleich"
- *: Verweis auf nachfolgende Bemerkungen

BDOS-Interface:

```
Eingang: CPU-Register C : BDOS-Funktionsnummer
         CPU-Register DE: Feldadressen, Vektoren
                       E : Eingabezeichen
         UP-Ansprung     : 0005H
Ausgang: CPU-Register A : Status, Zeichen
         CPU-Register HL: Vektoren, Adressen
```

Fussnoten:

```
*1: Sonderzeichen
    ^H: Rueckschritt
    CR: Wagenruecklauf
    LF: Zeilenschaltung
    ^I: Tabulation auf naechste 8. Stelle
    ^S: start/stop Konsolenausgabe
    ^P: List - Echo

*2: Status = 00 kein Zeichen im Konsoleneingabepuffer
           =/00 Zeichen im Konsoleneingabepuffer

*3: Steuerzeichen
    7F: Loeschen letztes Zeichen
    ^C: Sprung zum Warmstart, wenn als 1. Zeichen angeboten
    ^E: Ende der physischen Zeile
    ^H: Rueckschritt um 1 Zeichen
    ^J: (LF) Abschluss der Zeile
    ^M: (Abschlusstaste) Abschluss der Zeile
    ^R: Wiederausgabe der redigierten Konsolzelle
    ^U: ganze Zeile zurueckweisen
    ^X: Ruecksetzen an Zeilenanfang mit Loeschen

*4: Struktur des Puffers
    DE: +0 I +1 I +2 I +3 I ... I +n+1
        mx I nx I Z1 I Z2 I ... I  Zn

       mx - Kapazitaet des Puffers
       nx - Fuellstand des Puffers
        Z - Zeichen im Puffer
```

```
*5: Fehler-Code
    A = 00 : Operation ok
      =/00 : Operation nicht ausfuehrbar

*6: Direktzugriffs-Code
    A = 01 : Lesen ungeschriebener Daten
      = 03 : Extent-Wechsel nicht moeglich
      = 06 : Zugriff auf EOD-Satz der Datei


*7: Verzeichnis-Code
    A = FF       : Fehler
      = 0,1,2,3  : Nummer des Eintrages im aktuellen
                   Verzeichnis-Satz

*8: Aufbau des Dateibeschreibers fuer Umbenennung
    FCB  +0 - +15 : Datei alt (LW = 0,1, ... 16)
        +16 - +31 : Datei neu (LW = 0 oder LW-alt)

*9: Batchmode
    A = 00 : keine Batchmodedatei
    A = FF : Batchmodedatei vorhanden
```

Aufbau des FCB:

```
    +00 I    dr    I Laufwerkcodierung
        I          I dr = 0 : Bezugs-LW
        I          I      1 : LW A
        I          I     16 : LW P
    +01 I f1 - f8  I Dateiname in ASCII
        I          I Flags in den 8. Bits f1' - f8'
        I          I fuer Nutzerflags reserviert
    +09 I t1 - t3  I Dateityp in ASCII
        I          I Flags in den 8. Bits t1' - t3'
        I          I t1' = 1/0 : geschuetzte/ungeschuetzte Datei
        I          I t2' = 1/0 : SYSTEM-/Verzeichnis-Datei
        I          I t3'       : reserviert
    +12 I    ex    I laufende Nummer des Dateiteils.
        I          I (Extent) 0..31
    +13 I  s1 - 2  I reserviert fuer internen Gebrauch
    +15 I    rc    I Satz-zaehler innerhalb laufenden Extents
    +16 I d0 - 15  I Blockverzeichnis des Extents
    +32 I    cr    I laufende Satznummer der sequentiellen
        I          I Datei innerhalb des Extents
    +33 I  r0 - 2  I Satznummer der Datei fuer Direktzugriff
    +36 I          I Ende FCB
```

Aufbau der LW-Vektoren

```
Vektor im CPU-Doppelregister
bit 0  : LW A
    1  : LW B
    :  : :: :
   15  : LW P
```

4. Beispiel fuer die Anwendung der BDOS-Funktionen

Mit dem folgenden einfachen Programmbeispiel soll die prinzipielle Anwendung der BDOS-Funktionen demonstriert werden.

Aufgabe dieses Erfassungsprogramms ist:

- Bediener gibt Dateiname vor (Dateityp ist TEX). Eine eventuell bestehende Datei gleicher Bezeichnung wird geloescht!

- Dann kann er Saetze bis zu max. 126 Zeichen eingeben, die auf dem Bildschirm dargestellt werden. Gibt er weniger als 126 Zeichen ein, so hat er mit der Abschlusstaste die Eingabe zu beenden. Korrekturen vor Abschluss eines Satzes sind moeglich (entsprechend BDOS-Funktion #10).

- Jeder Satz, am Ende mit Space aufgefuellt und mit den Steuerzeichen Zeilenschaltung und Kursor an Zeilenanfang versehen, wird auf der Diskette im aktuellen Laufwerk als ein Satz mit 128 Byte in der vorgewaehlten Datei aufgezeichnet.

- Gibt der Bediener nur "END" bzw. "end" ein, so wird dies als letzter Satz gedeutet. Das Programm wird nach dem Aufzeichnen dieses Satzes und dem Schliessen der Datei beendet.

- Tritt ein Fehler auf, so wird die Datei nicht aufgezeichnet. Der Fehler wird angezeigt.

*** Beispiel BDOS-Funktionen ***

```
beispiel1        ASM 1520 (SCPX) V 1.0           1

                                                 title   beispiel1
0000'                        aseg
                             org     100h
                             .z80
                             page    36
```

```
beispiel1       ASM 1520 (SCPX) V 1.0           1-1


0005            bdos    equ     5               ;Adresse vom Sprung
                                                ;an BDOS
000D            cr      equ     0dh             ;Steuerzeichen:
                                                ;Kursor an
                                                ;Zeilenanfang
000A            lf      equ     0ah             ;Steuerzeichen:
                                                ;Zeilenschaltung
000C            home    equ     0ch             ;Steuerzeichen:
                                                ;Kursor an BS—Anf.

0100  31 0269   start:  ld      sp,stack+16     ;eigener Stack
0103  11 0269           ld      de,text         ;Programmanzeige
                                                ;auf Bildschirm
0106  0E 09             ld      c,9
0108  CD 0005           call    bdos

010B  11 0283           ld      de,name         ;Dateiname:
                                                ;Eingabeauf-
                                                ;forderung
010E  0E 09             ld      c,9
0110  CD 0005           call    bdos
0113  11 01B6           ld      de,puffer       ;Eingabe Dateiname
0116  0E 0A             ld      c,10
0118  CD 0005           call    bdos
011B  3A 01B7           ld      a,(puffer+1)    ;Anzahl Zeichen
                                                ;Dateiname
011E  4F                ld      c,a
011F  06 00             ld      b,0
0121  21 01B8           ld      hl,puffer+2
0124  11 0239           ld      de,fcb+1
0127  ED B0             ldir                    ;Dateiname--->FCB

0129  3E 7E             ld      a,126           ;max. Eing.laenge
                                                ;f.Satzeingabe
012B  32 01B6           ld      (puffer),a
012E  11 0238           ld      de,fcb          ;loeschen einer
                                                ;evt. bestehenden
                                                ;Datei
0131  D5                push    de              ;gleichen Namens
0132  0E 13             ld      c,19
0134  CD 0005           call    bdos
```

```
beispiel1       ASM 1520 (SCPX) V 1.0           1-2


0137  D1                pop     de              ;Einrichten Datei
0138  D5                push    de
0139  0E 16              ld      c,22
013B  CD 0005            call    bdos
013E  3C                inc     a
013F  28 64              jr      z,fehler        ;Sprung bei Fehler

0141  11 01B8            ld      de,dma          ;Adresse fuer FD-
                                                 ;Uebertragung
0144  0E 1A              ld      c,26
0146  CD 0005            call    bdos
0149  1E 0D      loop:   ld      e,cr            ;Kursoreinstellung
014B  0E 02              ld      c,2
014D  CD 0005            call    bdos

0150  1E 0A              ld      e,lf
0152  0E 02              ld      c,2
0154  CD 0005            call    5

0157  11 01B6            ld      de,puffer       ;Texteingabe
015A  0E 0A              ld      c,10
015C  CD 0005            call    bdos

015F  21 01B8            ld      hl,dma          ;Ausgabepuffer FD
0162  3A 01B7            ld      a,(puffer+1)    ;Anzahl Zeichen
0165  4F                 ld      c,a
0166  06 00              ld      b,0
0168  09                 add     hl,bc           ;erstes zu
                                                 ;loeschendes Byte
0169  3E 7E              ld      a,126           ;max. Anz. Zeichen
016B  91                 sub     c
016C  28 07              jr      z,write         ;bei 126 Zeichen
```

```
beispiel1       ASM 1520 (SCPX) V 1.0           1-3


016E  47                ld       b,a
016F  3E 20             ld      a,20h          ;Fuellzeichen
0171  77        loesch: ld      (hl),a
0172  23                inc     hl
0173  10 FC             djnz    loesch

0175  D1        write:  pop     de             ;Write sequentiell
0176  D5                push    de
0177  0E 15              ld      c,21
0179  CD 0005            call    bdos
017C  B7                or      a
017D  20 26              jr      nz,fehler      ;Sprung an Fehler-
                                                ;behandlung
017F  3A 01B7            ld      a,(puffer+1)   ;Abfrage auf "end"
                                                ;bzw "END"
0182  FE 03              cp      3
0184  20 C3              jr      nz,loop        ;keine 3 Zeichen
                                                ;eingegeben
0186  3A 01B8            ld      a,(puffer+2)
0189  CB AF              res     5,a
018B  FE 45              cp      'E'            ;Grossbuchstaben
                                                ;erzeugen
018D  20 BA              jr      nz,loop        ;Fortsetzen der
                                                ;Erfassung
018F  3A 01B9            ld      a,(puffer+3)
0192  CB AF              res     5,a
0194  FE 4E              cp      'N'
0196  20 B1              jr      nz,loop        ;Fortsetzen der
                                                ;Erfassung
0198, 3A 01BA            ld      a,(puffer+4)
019B  CB AF              res     5,a
019D  FE 44              cp      'D'
019F  20 A8              jr      nz,loop        ;Fortsetzen der
                                                ;Erfassung

01A1  0E 10              ld      c,16           ;Schliessen
                                                ;der Datei
01A3  18 0A              jr      ende
```

```
beispiel1       ASM 1520 (SCPX) V 1.0           1-4


01A5  11 0291   fehler: ld      de,fetext     ;Fehleranzeige
01A8  0E 09             ld      c,9
01AA  CD 0005           call    bdos

01AD  0E 13             ld      c,19          ;Loeschen Datei
                                              ;im Fehlerfall
01AF  D1        ende:   pop     de            ;FCB- Adresse
01B0  CD 0005           call    bdos
01B3  C3 0000           jp      0             ;Ende mit Warmstart

01B6  08 00     puffer: db      8,0           ;Eingabepuffer
                                              ;(max.Laenge,
                                              ;Istlaenge)
01B8            dma:    ds      126           ;Puffer f. Eingabe
                                              ;und FD-Ausgabe
0236  0D 0A             db      cr,lf         ;letzte konstante
                                              ;Zeichen f.
0238  00        fcb:    db      0             ;FCB:aktuelles
                                              ;Laufwerk
0239                    ds      8,20h         ;    fuer Dateiname
0241  54 45 58          db      'TEX'         ;    Dateityp
0244                    ds      21,0          ;    Restbyte
                                              ;    (auf 0
                                              ;    eingestellt)
0259            stack:  ds      16,0          ;Anwenderstack

0269  0C 20 20 20 text:   db      home,'    ERFASSUNGSPROGRAMM',
026D  20 45 52 46                 cr,lf,¤
0271  41 53 53 55
0275  4E 47 53 50
0279  52 4F 47 52
027D  41 4D 4D 0D
0281  0A 24
0283  0A 44 61 74 name:   db      lf,'Dateiname : ',¤
0287  65 69 6E 61
028B  6D 65 20 3A
028F  20 24
0291  0D 0A 0A 20 fetext: db      cr,lf,lf,' Fehler (Directory
oder
                                  Diskette voll) !'
0295  46 65 68 6C
0299  65 72 20 28
```

*** Beispiel BDOS-Funktionen ***

```
beispiel1        ASM 1520 (SCPX) V 1.0           1-5


029D  44 69 72 65
02A1  63 74 6F 72
02A5  79 20 6F 64
02A9  65 72 20 44
02AD  69 73 6B 65
02B1  74 74 65 20
02B5  76 6F 6C 6C
02B9  29 20 21
02BC  0D 0A 24            db      cr,lf,24h

                          end
```

*** Beispiel BDOS-Funktionen ***

```
beispiel1       ASM 1520 (SCPX) V 1.0           S

Macros:

Symbols:
0005    BDOS            000D    CR              01B8    DMA
01AF    ENDE            0238    FCB             01A5    FEHLER
0291    FETEXT          000C    HOME            000A    LF
0171    LOESCH          0149    LOOP            0283    NAME
01B6    PUFFER          0259    STACK           0100    START
0269    TEXT            0175    WRITE

No Fatal error(s)
```

5. Beschreibung des BIOS

5.1. Aufgabe des BIOS

Das BIOS ist der von der Hardware abhaengige Modul des Betriebssystems SCP. Es beinhaltet die fuer die spezielle Hardware notwendigen Ein-/Ausgaberoutinen. Damit bildet es die Schnittstelle zwischen der Hardware und dem hardwareunabhaengigen Teil des Betriebssystems bzw. dem Anwenderprogramm. Das BIOS kann mit Hilfe eines Installierprogrammes an eine veraenderte Hardware angepasst werden.

Die im BIOS enthaltenen Ein-/Ausgaberoutinen koennen in drei Gruppen zusammengefasst werden:

1. Systeminitialisierung
2. Zeichenein- und -ausgabe
3. Diskettenein- und -ausgabeoperationen

Die Routinen erreicht man ueber einen sogenannten "Sprungvektor". Der Sprungvektor steht am BIOS-Anfang und stellt eine zusammenhaengende Folge von Sprungbefehlen dar. Nachfolgend sind die Routinen der oben genannten Gruppen angegeben, zu denen je ein Sprungbefehl im Sprungvektor enthalten ist:

Systeminitialisierung:
- Kaltstartroutine
- Warmstartroutine

Zeichenein-/-ausgabeoperationen:
- Status CONSOLE-Geraet
- Eingabe von CONSOLE-Geraet
- Ausgabe auf CONSOLE-Geraet
- Ausgabe auf LIST-Geraet
- Status LIST-Geraet
- Ausgabe auf PUNCH-Geraet
- Eingabe von READER-Geraet

Diskettenein-/-ausgabeoperationen:
- Positionieren Spur Null
- Laufwerk auswaehlen
- Spur auswaehlen
- Transformation Sektornummer
- Sektor auswaehlen
- Datenpufferadresse setzen
- Selektierten Sektor lesen
- Schreiben selektierten Sektor

5.2. BIOS-Schnittstelle

Im Betriebssystem SCP hat der BIOS-Sprungvektor nachfolgend beschriebenen Aufbau. Die symbolischen Sprungadressen dienen nur zum besseren Verstaendnis der nachfolgenden Beschreibungen.

```
Sprungnummer      Befehl                Funktion
------------      ------                --------
     0          JMP BOOT       ; Kaltstartroutine
     1          JMP WBOOT      ; Warmstartroutine
     2          JMP CONST      ; CONSOLE-Status abfragen
     3          JMP CONIN      ; CONSOLE-Eingabe
     4          JMP CONOUT     ; CONSOLE-Ausgabe
     5          JMP LIST       ; LIST-Ausgabe
     6          JMP PUNCH      ; PUNCH-Ausgabe
     7          JMP READER     ; READER-Eingabe
     8          JMP HOME       ; Spur Null einstellen
     9          JMP SELDSK     ; Laufwerk auswaehlen
    10          JMP SETTRK     ; Spur auswaehlen
    11          JMP SETSEC     ; Sektor auswaehlen
    12          JMP SETDMA     ; Datenpufferadresse setzen
    13          JMP READ       ; Selektierten Sektor lesen
    14          JMP WRITE      ; Selektierten Sektor schreiben
    15          JMP LISTST     ; LIST-Status abfragen
    16          JMP SECTRAN    ; Umrechnen Sektornummer
```

Die angegebenen Routinen werden als Unterprogramme aufgerufen, enden also mit einem Ruecksprung (mit Ausnahme der Warm- und Kaltstartroutine, fuer die eigene Regeln gelten). Dabei werden eventuell benoetigte Werte in folgenden Prozessorregistern uebergeben:

- an das BIOS: 8-Bit-Werte in Register C,
  16-Bit-Werte in Registerpaar BC,
  (zweiter 16-Bit-Wert in Registerpaar DE);
- vom BIOS: 8-Bit-Werte in Register A,
  16-Bit-Werte in Registerpaar HL.

Ein Programm kann die BIOS-Routinen auch unmittelbar nutzen. Der Ausgangspunkt dazu ist der Sprung auf Adresse 0 . Hier befindet sich ein Sprung zur Warmstartroutine im BIOS-Teil (zweite Eintragung im Sprungvektor). Aus der Zieladresse dieses Sprungbefehls und der Nummer der benoetigten BIOS-Routine laesst sich leicht die Adresse berechnen, die das Programm gegebenenfalls aufrufen muss:

(<Sprungnummer> -1) x 3 +<Zieladresse des Sprungs auf Adresse 0>

Wenn ein Programm beispielsweise den Zustand des CONSOLE-Geraetes (Sprungnummer = 2) wissen moechte, dann kann das mit dem folgenden Unterprogramm geschehen:

```
LD  HL,0          ; HL loeschen
LD  DE,1          ; (Nummer der BIOS-Routine) - 1
ADD HL,DE         ; Abstand des Sprungs vom Anfang
ADD HL,DE         ; berechnen
ADD HL,DE
EX  DE,HL
LD  HL,(1)        ; Adresse Warmstart
ADD HL,DE         ; Adresse der BIOS-Routine
JP  (HL)          ; Sprung zur BIOS-Routine
```

Beachte: Dieses Unterprogramm kann nicht zur Berechnung der Adresse der Kaltstartroutine verwendet werden !

5.3. BIOS-Bestandteile

Im folgenden werden die drei Gruppen der Routinen im BIOS naeher betrachtet.

5.3.1. Initialisierung des SCP

Es gibt im BIOS zwei Routinen zur Initialisierung des Systems, den Kaltstart und den Warmstart.

Die Kaltstartroutine BOOT wird nur nach dem Urladen oder der Neuinstallation eines SCP im Speicher aktiviert. Sie fuehrt eine grundlegende Systeminitialisierung sowohl des Betriebssystems als auch der Hardware durch und gibt einen System-Kaltstarttext auf Bildschirm aus. Falls im SCP installiert wird zusaetzlich ein Nutzer-Kaltstarttext ausgegeben und ein Kaltstartkommando ausgefuehrt. Die Initialisierung schliesst mit der Uebergabe der Steuerung an das CCP ab.

Die Warmstartroutine WBOOT wird immer dann aktiviert, wenn ein Nutzerprogramm zur Adresse 0000H verzweigt. Nach der Initialisierung der Systemparameter wird zum CCP verzweigt.

5.3.2. Logische Ein-/Ausgabekanaele

Das System SCP unterstuetzt 4 Kanaele:

- einen Ein-/Ausgabekanal
- einen Eingabekanal
- zwei Ausgabekanaele.

Ueber die im Punkt 3. beschriebenen BDOS-Funktionen sind diese Kanaele ansprechbar.

Die Unterprogramme, hoehere Programmiersprachen und verschiedene Kommandos (z.B. PIP, STAT, ... siehe dazu auch Syst.Hb./Bediener) greifen ueber die BDOS-Funktionen auf die E/A-Kanaele zu.

Die Schluesselbegriffe CON:, LST:, PUN: und RDR: stellen die Kanalbezeichnungen dar, wie sie u.a. in den Programmen PIP und STAT Verwendung finden. Die 4 logischen E/A-Kanaele haben folgende Eigenschaften:

CON: (CONSOLE)
Dieser logische Kanal kann sowohl zur Ein- als auch zur Ausgabe von Zeichen genutzt werden.

Eingaberoutinen:
CONIN fuer zeichenweise Eingabe
CONST zur Abfrage, ob ein Zeichen bereit steht

Ausgaberoutinen:
CONOUT fuer zeichenweise Ausgabe

LST: (LIST)
Der logische Kanal dient zur Ausgabe einzelner Zeichen. Das LIST-Geraet ist normalerweise ein Drucker.

Eingaberoutinen:
LISTST fuer Statusabfrage des LIST-Geraetes

Ausgaberoutinen:
LIST fuer zeichenweise Ausgabe

PUN: (PUNCH)
Der logische Kanal dient zur zeichenweise Ausgabe.

Ausgaberoutinen:
PUNCH fuer zeichenweise Ausgabe

RDR: (READER)
Der logische Kanal dient zur zeichenweisen Eingabe.

Eingaberoutinen:
READER fuer zeichenweise Eingabe

Jedem logischen E/A-Kanal kann genau einer von jeweils 4 moeglichen Subkanaelen zugeordnet werden. Diese Zuordnung wird durch den Inhalt des IO-Bytes auf Adresse 0003H bestimmt.

Den Subkanaelen werden im BDOS bis zu 4 physische Geraetetreiber f}r Zeichenein-/-ausgabe zugeordnet.

Die durch das IO-Byte festgelegte Zuordnung des log. Kanals zum Subkanal kann durch BDOS-Funktionen abgefragt und/oder geaendert werden (siehe dazu 3.3.1.). Auf Kommandoebene kann eine solche Zustandsabfrage bzw. -aenderung des IO-Byte mit Stat erfolgen (siehe dazu auch Syst.-Hb./Bediener, Pkt.4.2.5.2.)

Die Zuordnung Subkanal zum Geraetetreiber ist mit dem Installationsprogramm "INSTSCP" abfragbar und auch aenderbar. Information ueber diese Zuordnung wird durch die BDOS-Funktion nicht geliefert.

Der Zusammenhang zwischen Kanal, Subkanal und Geraetetreiber, sowie eine Information ueber welchen Funktionen bzw. Programme auf diese Zuordnung Einflu- genommen werden kann, ist dem Anhang D zu entnehmen.

Der Aufbau des IO-Byte ist wie folgt:

		IO-Byte, Bit-#							
log. E/A-Kan.	Subkanal	7	6	5	4	3	2	1	0
CON:	TTY:	x	x	x	x	x	x	0	0
	CRT:	x						0	1
	BAT:	x						1	0
	UC1:	x						1	1
RDR:	TTY:	x				0	0		x
	PTR:	x				0	1		x
	UR1:	x				1	0		x
	UR2:	x				1	1		x
PUN:	TTY:	x		0	1				x
	PTP:	x		0	1				x
	UP1:	x		1	1				x
	UP2:	x		1	1				x
LST:	TTY:	0	0						x
	CRT:	0	1						x
	LPT:	1	0						x
	UL1:	1	1	x	x	x	x	x	x

Die Subkanaele gleichen Namens bei verschiedenen log. E/A-Kanaelen koennen unterschiedliche Bedeutung haben. So koennen z.B. unter TTY: bei CON:, RDR:, PUN:, und LST voellig unterschiedliche Geraetetreiber angesprochen werden, je nachdem welche Treiber durch "INSTSCP" dem jeweiligen TTY zugeordnet worden sind (siehe Installationshandbuch Pkt. 7.ff)

5.3.3. Diskettenarbeit

Die Arbeit mit den Disketten gliedert sich in die Laufwerksteuerung und den Datenverkehr.

5.3.3.1. Laufwerksteuerung

Die Laufwerksteuerung umfasst drei Schritte, die zum Adressieren eines Sektors auf der Diskette notwendig sind. Mit Sektor wird im weiteren ein 128-Byte grosser Aufzeichnungsabschnitt auf der Diskette bezeichnet.

1. Auswahl des gewuenschten Laufwerkes
   mittels der Routine SELDSK

2. Schreib-Lesekopf auf die Spur setzen, in der sich
   die Information befindet.
   mittels der Routine SETTRK

3. Adressierung des Sektors, der die Information enthaelt
   mittels der Routinen SECTRAN und SETSEC
   Das Einstellen der Sektornummer erfolgt in zwei Schritten. Im ersten Schritt erfolgt ueber die Routine SECTRAN die Umwandlung der logischen Sektornummer in die physische Sektornummer. Diese Umwandlung ermoeglicht eine Diskettenorganisation, die einen zeitoptimalen Zugriff auf die gewuenschte Information gewaehrleistet.
   Im zweiten Schritt wird ueber die Routine SETSEC die Adressierung des physischen Sektors vorgenommen.

5.3.3.2. Datenverkehr

Der Datenverkehr umfasst neben dem Lesen und Schreiben von Informationen weiterhin die Festlegung der Adresse des Datenpuffers, jenes 128-Byte-Bereiches im Speicher, der die Daten von der Diskette uebernimmt bzw. von dem sie kommen.
Liegt der Ort der Aufzeichnung auf der Diskette fest, und ist der Ort des Datenpuffers im Speicher bestimmt, dann koennen die Daten gelesen oder auf die Diskette geschrieben werden. Dazu bietet das BIOS die Routinen:

- SETDMA
  Festlegen des Datenpuffers als Ziel oder Herkunft der Daten

- READ
  Lesen eines 128-Byte-Sektors von der Diskette und Uebertragen in den Datenpuffer

- WRITE
  Schreiben der im Datenpuffer vorliegenden Information in den adressierten 128—Byte—Sektor

5.4. BIOS-Eintrittspunkte

Nachfolgend werden die zu den BIOS-Eintrittspunkten gehoerenden Routinen beschrieben.

BOOT Kaltstart
Aufrufparameter: —
Rueckkehrparameter: -
Die Routine erhaelt die Steuerung vom Urlader und ist verantwortlich fuer die grundlegende System-initialisierung. Sie gibt eine Kaltstartmeldung aus, die u. a. die Versionsnummer und das Erstellungsdatum des Betriebssystems sowie die Groesse des TPA enthaelt.
Die Kaltstartroutine kann durch das Installierprogramm so modifiziert werden, dass zusaetzlich noch ein Nutzer-Kaltstarttext ausgegeben wird.
Die Initialisierung schliesst mit der Uebergabe der Steuerung an das CCP ab, wobei je nach Installation ein Kaltstartkommando mit uebergeben werden kann.
Die Systemparameter werden wie folgt initialisiert:

Speicheradresse	Inhalt
0, 1, 2	Sprung zu WBOOT fuer Warmstart
3	Initialwert IOBYTE = 00H Der Wert kann durch das Programm INSTSCP manipuliert werden.
4	Bit 0 bis 3: Nummer des aktuellen Laufwerkes (0 = A,... , 15 = P) Bit 4 bis 7: Nummer des Nutzer-bereiches Es werden das Laufwerk A und Nutzerbereich 0 initialisiert. Initialwert = 00H
5, 6, 7	Sprung zum SCP - Anfang Zieladresse des Sprungs gibt das Ende des TPA an

Die Initialwerte fuer den Inhalt der Speicheradresen 1,2,6 und 7 sind von der Groesse des BIOS abhaengig.

WBOOT Warmstart
Aufrufparameter: -
Rueckkehrparameter: -

Die Systemparameter werden wie folgt initialisiert:

Speicheradresse	Inhalt
0, 1, 2	Sprung zu WBOOT fuer Warmstart
5, 6, 7	Sprung zum SCP - Anfang Zieladresse des Sprungs gibt das Ende des TPA an

Der Inhalt der Speicheradressen 3 und 4 bleibt unveraendert.
Nach vollstaendiger Initialisierung verzweigt die Routine fuer den Systemneustart zum CCP.

CONST Abfrage Status Kanal CON:
Aufrufparameter: -
Rueckkehrparameter: A - Status

Diese Routine untersucht das dem Kanal CON: zugewiesene Geraet und liefert:
im Register A = 0 falls kein Zeichen bereitsteht
A = 0FFH sonst.

CONIN Empfangen Zeichen von Kanal CON:
Aufrufparameter: -
Rueckkehrparameter: A - Empfangenes Zeichen

Diese Routine liest das naechste Zeichen von CON: in das Register A. Steht kein Zeichen bereit, wird bis zur Zeicheneingabe gewartet.

CONOUT Senden Zeichen auf Kanal CON:
Aufrufparameter: C - Sende-Zeichen
Rueckkehrparameter: -

Das in Register C bereitgestellte Zeichen wird auf Kanal CON: ausgegeben.

LIST Senden Zeichen auf Kanal LST:
Aufrufparameter: C - Sende-Zeichen
Rueckkehrparameter: —

Das in Register C bereitgestellte Zeichen wird auf den Kanal LST: ausgegeben.

PUNCH Senden Zeichen auf Kanal PUN:
Aufrufparameter: C - Sende-Zeichen
Rueckkehrparameter: -

Die Routine gibt das in Register C bereitgestellte Zeichen auf den Kanal PUN: aus.

READER Empfangen Zeichen von Kanal RDR:
Aufrufparameter: —
Rueckkehrparameter: A - Empfangenes Zeichen

Die Routine liest das naechste Zeichen von Kanal RDR: in Register A.
Die EOF-Bedingung wird durch 1AH (CTRL Z) geliefert.

HOME Positionieren Spur 0
Aufrufparameter: -
Rueckkehrparameter: -

Die Routine positioniert den Kopf des selektierten Laufwerkes auf die Spur Null.

SELDSK Selektieren Laufwerk
Aufrufparameter: C - Laufwerknummer
Rueckkehrparameter: HL - Diskettenparameterkopf DPH

Die Routine waehlt das im Register C angegebene Laufwerk fuer den naechsten Datenzugriff aus. Dabei enthaelt das Register C eine 0 fuer Laufwerk A, eine 1 fuer Laufwerk B, eine 2 fuer Laufwerk C bzw. eine 3 fuer Laufwerk D.
Bei jedem Aufruf der Routine wird in HL die Adresse eines Speicherbereiches bereitgestellt, in dem die Merkmale der betreffenden Diskette enthalten sind und Platz fuer Zwischenergebnisse beim Diskettenzugriff vorhanden ist. Dieser Speicherbereich, als Vorspann fuer die Diskettenparameter (Disk Parameter Header) bezeichnet und mit DPH abgekuerzt, ist fuer die BDOS-Arbeit vor allem deshalb wichtig, da hierueber die Informationen zur Diskettenstruktur und andere wichtige Daten erreicht werden. Der Inhalt des DPH-Bereiches wird spaeter beschrieben.
Wird SELDSK fuer ein nicht vorhandenes Laufwerk aufgerufen, dann wird in HL 0000H zurueckgegeben.
Die physische Laufwerksauswahl wird erst bei einem tatsaechlichen Datenzugriff (READ oder WRITE) ausgefuehrt.
Das niederwertigste Bit des Registers E ist 0, wenn es sich um den ersten Aufruf der Routine nach einem Kalt- oder Warmstart handelt.
Das selektierte Laufwerk wird erst durch erneuten Aufruf der Routine SELDSK geaendert.

SETTRK Spurpositionierung
Aufrufparameter: BC - Spurnummer
Rueckkehrparameter: —

Die Routine nimmt die Spurpositionierung entsprechend dem Inhalt des Registerpaares BC fuer den naechsten Zugriff auf das ausgewaehlte Laufwerk vor.
Im BIOS wird lediglich die Spurnummer gemerkt und die eigentliche Positionierung bis zu einem Lese- oder Schreibbefehl zurueckgestellt. Dadurch lassen sich Verwaltungsaufgaben vereinfachen. Wichtig ist lediglich, dass beim Datenzugriff der Kopf auf der richtigen Spur steht.
Die Spuradresse bleibt bis zum erneuten Aufruf der Routine erhalten.

SETSEC Sektorpositionierung
Aufrufparameter: BC — Sektornummer
Rueckkehrparameter: —
Entsprechend der im Registerpaar BC eingestellten Sektornummer wird auf dem ausgewaehlten Laufwerk der Sektor fuer den naechsten Diskettenzugriff positioniert.
Die Sektornummer wird zunaechst intern gemerkt und der eigentliche Zugriff bis zu einem Schreib- oder Lesebefehl verschoben.
Die Sektoradresse bleibt erhalten bis zu einem erneuten Aufruf der Routine.

SETDMA Einstellen Pufferadresse
Aufrufparameter: BC - Pufferadresse
Rueckkehrparameter: -

Die Routine erhaelt in BC die Anfangsadresse eines 128 Bytes umfassenden Speicherbereichs, der als Datenpuffer fuer alle nachfolgenden Schreib- und Leseoperationen dient.
Vom Betriebssystem wird standardmaessig ein Datenpuffer der Laenge 128 Bytes auf der Adresse 8ØH angelegt.

READ Lesen Sektor
Aufrufparameter: -
Rueckkehrparameter: A - Fehlerkode

Unter der Voraussetzung, dass Laufwerk, Spur, Sektor und Adresse des Datenpuffers festgelegt wurden, versucht die Routine den durch diese Parameter bestimmten Sektor zu lesen.
Im Register A wird folgender Fehlerkode zurueckgegeben.

Ø	fehlerfreies Lesen
1	Lesen nicht moeglich

Ist der Wert im Register A gleich Ø, dann wird vom Betriebssystem der Diskettenzugriff als erfolgreich abgeschlossen angenommen. Tritt jedoch ein Fehler auf, dann versucht das BIOS durch mehrmaliges Wiederholen festzustellen, ob der Fehler behebbar ist.
Bei Fehler wird von BIOS folgende Meldung ausgegeben:

```
BIOS ERR ON: x  T: yy  S: zz
```

Dabei bedeuten:

x	-	Laufwerk	(A,...)
yy	-	Spur	(ØØ,...)
zz	-	Sektor	(Ø1,...)

Nur wenn bei der SCP - Installation BIOS ERROR RESPONSE ON spezifiziert wurde, wird zusaetzlich folgender Quittungstext ausgegeben:

```
R - REPEAT, I - IGNORE, ^C - CANCEL
```

Der Bediener hat damit drei Moeglichkeiten, den Fehler zu quittieren:

R	— wiederholen der fehlerhaften Operation
I	- ignorieren des Fehlers und Rueckkehr mit Fehler in Register A
^C	- Abbruch des Programms und Warmstart.

WRITE Schreiben Sektor
Aufrufparameter: -
Rueckkehrparameter: A - Fehlerkode

Die Routine schreibt die Daten aus dem vorher festgelegten Datenpuffer auf das zuvor ausgewaehlte Laufwerk, Spur und Sektor. Die in Register A zurueckgegebenen Fehlerbedingungen und die Verfahrensweise bei Auftreten eines Fehlers sind analog der READ-Routine.

LISTST Abfrage des Status von Kanal LST:
Aufrufparameter: -
Rueckkehrparameter: A - Status Kanal LST:

Die Routine uebermittelt dem Nutzer eine Status-

Information ueber den Kanal LST:

Register A = 0FFH, wenn der Kanal LST: bereit ist, ein Zeichen zu uebernehmen.
Register A = 0 sonst.

SECTRAN Umwandeln der Sektornummer
Aufrufparameter: BC - umzuwandelnde Sektornummer (00,...)
DE - Adresse Umwandlungstabelle
Rueckkehrparameter: HL - umgewandelte Sektornummer

Die Routine erhaelt die logische Sektornummer in Registerpaar BC und die Adresse einer Umwandlungstabelle in Registerpaar DE.
Die logische Sektornummer (relativ zu Null angegeben) wird als ein Index in der Umwandlungstabelle verwendet.
Die durch die Umwandlung bestimmte physische Sektornummer wird im Registerpaar HL zurueckgegeben.
Hier wird noch nichts darueber ausgesagt, ob ueberhaupt ein und wenn ja welcher Sektor beim naechsten Datenzugriff tatsaechlich gelesen oder geschrieben wird. Es wird lediglich die Sektornummer bestimmt.

5.5. Beschreibung der Disketten

Aufgrund der Vielfalt von Diskettenlaufwerken und Diskettenformaten schliesst das BIOS die Moeglichkeit der Anpassung an verschiedene Laufwerke und Diskettenformate ein.
Deshalb enthaelt BIOS Tabellen, die dem Nutzer die Disketten- und Laufwerkseigenschaften mitteilen.

5.5.1. Diskettenparameterkopf DPH

Jedem Laufwerk ist ein 16 Byte grosser Diskettenparameterkopf (DPH — Disk Parameter Header) zugeordnet, der Informationen ueber das Diskettenlaufwerk enthaelt und Arbeitsbereiche fuer bestimmte BDOS-Operationen einschliesst.
Durch die BIOS-Routine SELDSK wird das Laufwerk ausgewaehlt und ausserdem die Adresse des zugehoerigen DPH im Registerpaar HL zurueckgegeben.

Ein DPH hat folgenden Aufbau:

Byte	Name	Bedeutung
0,1	XLT	Adresse der Uebersetzungstabelle fuer die Sektornummer Ist die Adresse gleich 0, dann stimmen logische und physische Sektornummer ueberein.
2,7		Arbeitsbereiche fuer BDOS, reserviert
8,9	DIRBUF	Adresse eines 128-Byte-Verzeichnispuffers alle DPH enthalten die gleiche Adresse
10,11	DPB	Adresse des Diskettenparameterblockes DPB) Jedes Laufwerk hat einen eigenen DPB.
12,13	CSV	Adresse eines Puffers, der fuer das Speichern eines Pruefsummenvektors zur Pruefung auf Diskettenwechsel erforderlich ist. Jedes Laufwerk hat einen eigenen Puffer.
14,15	ALV	Adresse eines Vektors, der die Disketten—belegung widerspiegelt. Bit n gleich 1 des Vektors bedeutet, dass der Block n der Diskette von einer Datei belegt ist. Bit n gleich 0 bedeutet, dass der Block unbelegt ist. Die ersten Bloecke und damit die ersten Bits sind durch das Verzeichnis belegt. Jedes Laufwerk hat einen eigenen Vektor.

Die fuer die verschiedenen Laufwerke zustaendigen DPH stehen lueckenlos hintereinander.
Die im DPH erfassten Daten und Speicherbereiche werden fuer jedes Laufwerk getrennt bereitgestellt.
Eine Ausnahme ist der 128-Byte-Puffer fuer die Verzeichnisauswertung. Er kann nur einmal im System vorhanden sein, da das BDOS immer nur ein Laufwerk zur Zeit erfassen kann und bei jeder Laufwerksumschaltung das Verzeichnis neu abfragt.

5.5.2. Diskettenparameterblock DPB

Der Diskettenparameterblock (DPB) fuer jedes Laufwerk ist wesentlich umfangreicher. In diesem Block sind alle Informationen zusammengefasst, die zur Verwaltung der betreffenden Diskette notwendig sind.

Dies umfasst:

- Informationen zur Speicherkapazitaet und
- Informationen zur Speicherorganisation.

Der DPB enthaelt unter anderem:
- Angaben zur Anzahl von Sektoren pro Spur,
- Angaben zur Anzahl von Sektoren pro Block,
- Angaben zur Groesse und Lage des Verzeichnisses sowie dazu, ob die Verzeichniseintraege bei jedem Zugriff auf Diskettenwechsel ueberprueft werden sollen und schliesslich
- eine Angabe zur Anzahl der auf der betreffenden Diskette fuer das Betriebssystem reservierten Spuren.

Diese Informationen sind im DPB wie folgt festgehalten:

Byte	Name	Bedeutung
0,1	SPT	Sektoren pro Spur
2	BSH	Blockverschiebungsfaktor darin ist die Blockgroesse verschluesselt als LOG (<Blockgroesse>/128) 2 Dieser Wert stellt ein Mass fuer die Anzahl der Sektoren pro Block dar.
3	BLM	Blockmaske wiederspiegelt ebenfalls die Blockgroesse Fuer die Blockmaske gilt: (2 ** BSH) - 1

Zwischen Blockgroesse, Blockverschiebungsfaktor und Blockmaske bestehen folgende feste Beziehungen:

Blockgroesse	BSH	BLM
1024	3	7
2048	4	15
4096	5	31
8192	6	63
16384	7	127

EXM — Extentmaske
ist definiert durch die Blockgroesse und die Anzahl der Bloecke pro Diskette
Ihre Groesse haengt von der Organisation des Verzeichniseintrages ab. Dieser enthaelt als wesentlichsten Teil fuer die Speicherverwaltung die Nummern der jeweils belegten Bloecke:
16 Eintraege zu je 1 Byte bei weniger als 256 Bloecken pro Diskette oder
8 Eintraege zu je 2 Byte bei mehr als 255 Bloecken pro Diskette.
Im einzelnen bestehen die Beziehungen:

Blockgroesse	Extentmaske fuer mehr als 255 Bloecken	Extentmaske fuer weniger als 256 Bloecken
1024	---	0
2048	0	1
4096	1	3
8192	3	7
16384	7	15

5,6 DSM — Anzahl der Bloecke pro Diskette minus 1 (einschliesslich des Verzeichnisses, aber ohne Systemspuren)

7,8 DRM — Anzahl der Verzeichniseintragungen minus 1
Die Groesse einer Verzeichniseintragung betraegt 32 Byte.

9,10	AL0,AL1	16-Bit-Vektor, in dem die vom Verzeichnis belegten Bloecke vermerkt sind Dieser Vektor wird beim ersten Laufwerkszugriff an den Anfang der Belegungstabelle kopiert und dient so zur Reservierung der Verzeichnisbloecke. Er ist aus diesem Grund umgekehrt als sonst ueblich organisiert: Die Zaehlung beginnt mit dem hoechstwertigen Bit, so hat der Vektor beispielsweise bei vier Verzeichnisbloecken den Wert 0F000H.
11,12	CKS	Groesse des Verzeichnis-Pruefvektors (Anzahl zu pruefender Verzeichniseintragungen dividiert durch 4)
13,14	OFF	Anzahl der Systemspuren

Die auf OFF folgenden 20 Byte sind fuer die interne Steuerung der verschiedenen physischen Laufwerke erforderlich und duerfen nicht geaendert werden.

Anhang A

Reservierte Speicherplaetze

Die Speicherplaetze zwischen 0000H und 00FFH werden als Verstaendigungsbereich bezeichnet enthalten verschiedene vom Betriebssystem benutzte Daten und Anweisungen. Die Anweisungen und Daten werden im folgenden dargestellt.

Speicherplaetze von	bis	Inhalt
0000H	0002H	Sprung zum BIOS-Eintrittspunkt WBOOT Damit ist ein einfacher programmierter Neustart (Sprung zu Adresse 0) moeglich.
0003H		Enthaelt das bereits beschriebene IOBYTE
0004H		Nummer des aktuellen Laufwerkes und Benutzernummer
0005H	0007H	Enthaelt eine Sprunganweisung zum BDOS Die Sprunganweisung liefert einmal den Haupteintrittspunkt in das BDOS und zum anderen stellt die Sprungadresse die niedrigste vom Betriebssystem verwendete Speicheradresse dar. Debugger veraendern die Sprungadresse, um den durch sie reduzierten Speicher zu kennzeichnen.
0008H	002FH	RST 1 bis RST 5 von SCP nicht verwendet
0030H	0037H	RST 6 fuer SCP reserviert, derzeit nicht benutzt
0038H	003AH	RST 7 enthaelt waehrend Debug-Mode eine Spunganweisung in den Debugger fuer progammierte Haltepunkte, wird vom Betriebssystem aber nicht benutzt
003BH	005EH	reserviert
005CH	007FH	durch CCP erzeugter Standard-FCB
0080H	00FFH	Standard-Datenpuffer

Anhang B

Bildschirmsteuerzeichen B e f e h l s a t z 1

Hex. code	Dez. code	Wirkung des Kommandos
01	1	Setzen Kursor auf linke obere Ecke (HOME Position)
07	7	ohne Wirkung
08	8	Setzen Kursor um eine Position nach links
0A	10	Setzen Kursor um eine Zeile nach unten
0C	12	Bildschirm loeschen und Kursor auf linke obere Ecke setzen
0D	13	Setzen Kursor auf Zeilenanfang
0E	14	Umschalten in den ersten Zeichensatz
0F	15	Umschalten in den zweiten Zeichensatz
14	20	Loeschen ab Kursorposition bis Bildschirmende
15	21	Setzen Kursor um eine Position nach rechts
16	22	Loeschen ab Kursorposition bis Zeilenende
18	24	Loeschen Kursorzeile und setzen Kursor an den Zeilenanfang
1A	26	Setzen Kursor um eine Zeile nach oben
1B	27	Einleiten der direkten Kursorpos. ESC , Zeile+80H , Spalte+80H

7F	127	Loeschen Zeichen auf Kursorpos. und Kursor um eine Pos. nach links
82	130	Kursor einschalten
83	131	Kursor ausschalten

Nachfolgende Steuerzeichen belegen 1 Byte im Hintergrundspeicher des Bildschirmes, d.h. diese Steuerzeichen sind fuer eine Berechnung der Kursorposition zu beruecksichtigen !!!

84	132	normale Zeichendarstellung
85	133	inverse Zeichendarstellung
86	134	intensive Zeichendarstellung
87	135	Zeichendarstellung invers und intensiv

F E L D A T T R I B U T E

Das Feldattribut belegt 1 Byte im Hinergrundspeioher!

ESC-Folge	Zeichendarstellung
1B 5E 40	normal
1B 5E 41	intensiv
1B 5E 42	blinkend
1B 5E 43	blinkend und intensiv

1B 5E 44	Zeichen aus alternativem Code-PROM
1B 5B 45	Zeichen aus alternativem Code-PROM intensiv
1B 5E 46	Zeichen aus alternativem Code-PROM blinkend
1B 5E 47	Zeichen aus alternativem Code-PROM blinkend und intensiv
1B 5E 50	invers
1B 5E 51	invers und intensiv
1B 5E 52	invers und blinkend
1B 5E 53	invers, blinkend und intensiv
1B 5E 54	Zeichen aus alternativem Code-PROM invers
1B 5E 55	Zeichen aus alternativem Code-PROM invers und intensiv
1B 5E 56	Zeichen aus alternativem Code-PROM invers und blinkend
1B 5E 57	Zeichen aus alternativem Code-PROM invers, blinkend und intensiv

P S E U D O G R A F I K S Y M B O L E

ESC-Folge	Grafiksymbole Darstellungsweise	Bild
1B 5F 40	linke obere Ecke normal	
1B 5F 41	linke obere Ecke intensiv	
1B 5F 42	linke obere Ecke blinkend	
1B 5F 43	linke obere Ecke intensiv und blinkend	
1B 5F 44	rechte obere Ecke normal	
1B 5F 45	rechte obere Ecke intensiv	
1B 5F 46	rechte obere Ecke blinkend	
1B 5F 47	rechte obere Ecke intensiv und blinkend	
1B 5F 48	linke untere Ecke normal	
1B 5F 49	linke untere Ecke intensiv	
1B 5F 4A	linke untere Ecke blinkend	

1B 5F 4B	linke untere Ecke intensiv und blinkend	
1B 5F 4C	rechte untere Ecke normal	
1B 5F 4D	rechte untere Ecke intensiv	
1B 5F 4E	rechte untere Ecke blinkend	
1B 5F 4F	rechte untere Ecke intensiv und blinkend	
1B 5F 50	oberer Intersect normal	
1B 5F 51	oberer Intersect intensiv	
1B 5F 52	oberer Intersect blinkend	
1B 5F 53	oberer Intersect intensiv und blinkend	
1B 5F 54	rechter Intersect normal	
1B 5F 55	rechter Intersect intensiv	
1B 5F 56	rechter Intersect blinkend	

1B 5F 57	rechter Intersect intensiv und blinkend	
1B 5F 58	linker Intersect normal	
1B 5F 59	linker Intersect intensiv	
1B 5F 5A	linker Intersect blinkend	
1B 5F 5B	linker Intersect intensiv und blinkend	
1B 5F 5C	unterer Intersect normal	
1B 5F 5D	unterer Intersect intensiv	
1B 5F 5E	unterer Intersect blinkend	
1B 5F 5F	unterer Intersect intensiv und blinkend	
1B 5F 60	horizontale Linie normal	
1B 5F 61	horizontale Linie intensiv	
1B 5F 62	horizontale Linie blinkend	

1B 5F 63	horizontale Linie intensiv und blinkend	-----
1B 5F 64	vertikale Linie normal	I
1B 5F 65	vertikale Linie intensiv	I
1B 5F 66	vertikale Linie blinkend	I
1B 5F 67	vertikale Linie intensiv und blinkend	I
1B 5F 68	Kreuz normal	--+--
1B 5F 69	Kreuz intensiv	--+--
1B 5F 6A	Kreuz blinkend	--+--
1B 5F 6B	Kreuz intensiv und blinkend	--+--

S O N D E R B E F E H L E

Es wird k e i n Speicherplatz im Hintergrundspeicher des Bildschirmes belegt !!

ESC-Folge	Bedeutung
1B 5F 30	Code-PROM 0 aktivieren
1B 5F 31	Code-PROM 1 aktivieren

Anhang C

BDOS - Funktionen

Nr.	Bezeichnung	Eingang	Ausgang
0	Warmstart — — —		— — —
1	Konsoleneingabe + Echo	— — —	A: Zeichen (*1)
2	Konsolenausgabe	E: Zeichen (mit ^S, ^P)	- - -
3	Sequentielle Eingabe	- - -	A: Zeichen
4	Sequentielle Ausgabe	E: Zeichen	— — —
5	List - Ausgabe	E: Zeichen	— - -
6	direkte Konsolen E/A	E = FF: Status	A: Konsolen-Status (*2)
		E = /FF: Ausgabe Zeichen	- - -
7	IOBYTE abfragen	- - -	A: IOBYTE
8	IOBYTE belegen	E: IOBYTE	- - -
9	Zeichenkette ausgeben	DE —> Kette EKZ=¤, mit ^S,^P	— — —
10	Eingabe Konsolpuffer	DE -> Puffer (*3)	- - -
11	Konsolenstatus	— — —	A: Konsolen-Status (*2)
12	Version ermitteln	— — —	HL: Version
13	Diskettensystem zuruecksetzen	- - -	A: Batchmode — Flag (*9)
14	Auswahl Bezugs-LW	E: LW# (0...F)	- - -
15	Datei eroeffnen	DE -> FCB	A: DC(*7, =FF:keine Datei)
16	Datei schliessen	DE -> FCB	A: DC(*7, =FF:keine Datei)
17	erste Eintragung suchen	DE -> FCB	A: DC(*7, =FF:keine Datei)
18	folgende Eintragung suchen	- - -	A: DC(*7, =FF:keine Datei)
19	Dateien loeschen	DE -> FCB	A: DC(*7, =FF:keine Datei)
20	naechsten Satz lesen	DE -> FCB	A: EC(*5, =/00: EOF)
21	naechsten Satz schreiben	DE -> FCB	A: EC(*5, =/00:Disk, voll)
22	Datei erzeugen	DE -> FCB	A: DC(*7, =FF: Verz. voll)
23	Datei umbenennen	DE -> FCB (*8)	A: DC(*7, =FF:keine Datei)
24	Abfrage ange-schlossener LW	- - -	HL: LW - Vektor
25	Abfrage Bezugs-LW	- - -	A: LW# (0 - 15)

Nr.	Bezeichnung	Eingang	Ausgang
26	Datenpuffer adressieren	DE -> Puffer	- - -
27	Belegungstabelle (ALLOC) ermitteln	- - -	HL -> ALLOC
28	Schutz des Bezugs-LW	- - -	- - -
29	Abfrage geschuetzte LW	- - -	HL: LW - Vektor
30	Dateimerkmale setzen	DE -> FCB	A: DC(*7, =FF:keine Datei)
31	Diskettenparameterta belle (DPB) ermitteln	- - — -	HL -> DPB des selekt. LW
32	Nutzernummer abfragen/setzen	E= FF: Abfrage E= /FF: Setzen	A:Nutzer# (0-15)
33	direkt adressierten Satz lesen	DE -> FCB	A: RC(*6)
34	direkt adressierten Satz schreiben	DE -> FCB	A: RC(*6)
35	Dateigroesse berechnen	DE -> FCB	- - -
36	Berechnung der aktuellen Satzadresse	DE -> FCB	Eintrag in FCB + 33,34,35: r0-2=fkt(ex,cr)
37	ausgewaehlte Laufwerke zuruecksetzen	DE: LW-Vektor	A: 00
40	direkt adressierten Satz schreiben und Block initialisieren	DE -> FCB	A: RC(*6)

Erlaeuterungen:

- Die Zeichen "->" sind zu interpretieren als "Zeiger auf"
- Die Zeichen "=/" sind zu interpretieren als "ungleich"
- *: Verweis auf nachfolgende Bemerkungen

BDOS-Interface:

```
Eingang: CPU-Register C : BDOS-Funktionnummer
         CPU-Register DE: Feldadressen, Vektoren
                      E : Eingabezeichen
         UP-Ansprung    : 0005H
Ausgang: CPU-Register A : Status, Zeichen
         CPU-Register HL: Vektoren, Adressen
```

Fussnoten:

*1: Sonderzeichen
    ^H: Rueckschritt
    CR: Wagenruecklauf
    LP: Zeilenschaltung
    ^I: Tabulation auf naechste 8. Stelle
    ^S: start/stop Konsolenausgabe
    ^P: List - Echo

*2: Status = 00 kein Zeichen im Konsoleneingabepuffer
           =/00 Zeichen im Konsoleneingabepuffer

*3: Steuerzeichen
    7F: Loeschen letztes Zeichen
    ^C: Sprung zum Warmstart, wenn als 1. Zeichen angeboten
    ^E: Ende der physischen Zeile
    ^H: Rueckschritt um 1 Zeichen
    ^J: (LF) Abschluss der Zeile
    ^M: (Abschlusstaste) Abschluss der Zeile
    ^R: Wiederausgabe der redigierten Konsolzelle
    ^U: ganze Zeile zurueckweisen
    ^X: Ruecksetzen an Zeilenanfang mit Loeschen

*4: Struktur des Puffers
    DE: +0 I +1 I +2 I +3 I ... I +n+1
        mx I nx I Z1 I Z2 I ... I Zn

       mx - Kapazitaet des Puffers
       nx - Fuellstand des Puffers
        Z - Zeichen im Puffer

*5: Fehler-Code
    A = 00 : Operation ok
      =/00 : Operation nicht ausfuehrbar

*6: Direktzugriffs-Code
    A = 01 : Lesen ungeschriebener Daten
      = 03 : Extent-Wechsel nicht moeglich
      = 06 : Zugriff auf EOD-Satz der Datei

*7: Verzeichnis-Code
    A = FF   : Fehler
      = 01,2,3 : Nummer des Eintrages im aktuellen
                 Verzeichnis-Satz

*8: Aufbau des Dateibeschreibers fuer Umbenennung
    FCB +0 - +15 : Datei alt (LW = 0,1, ... 16)
        +16 - +31 : Datei neu (LW = 0 oder LW-alt)

*9: Batchmode
    A = 00 : keine Batchmodedatei
    A = FF : Batchmodedatei vorhanden

Aufbau des FCB:

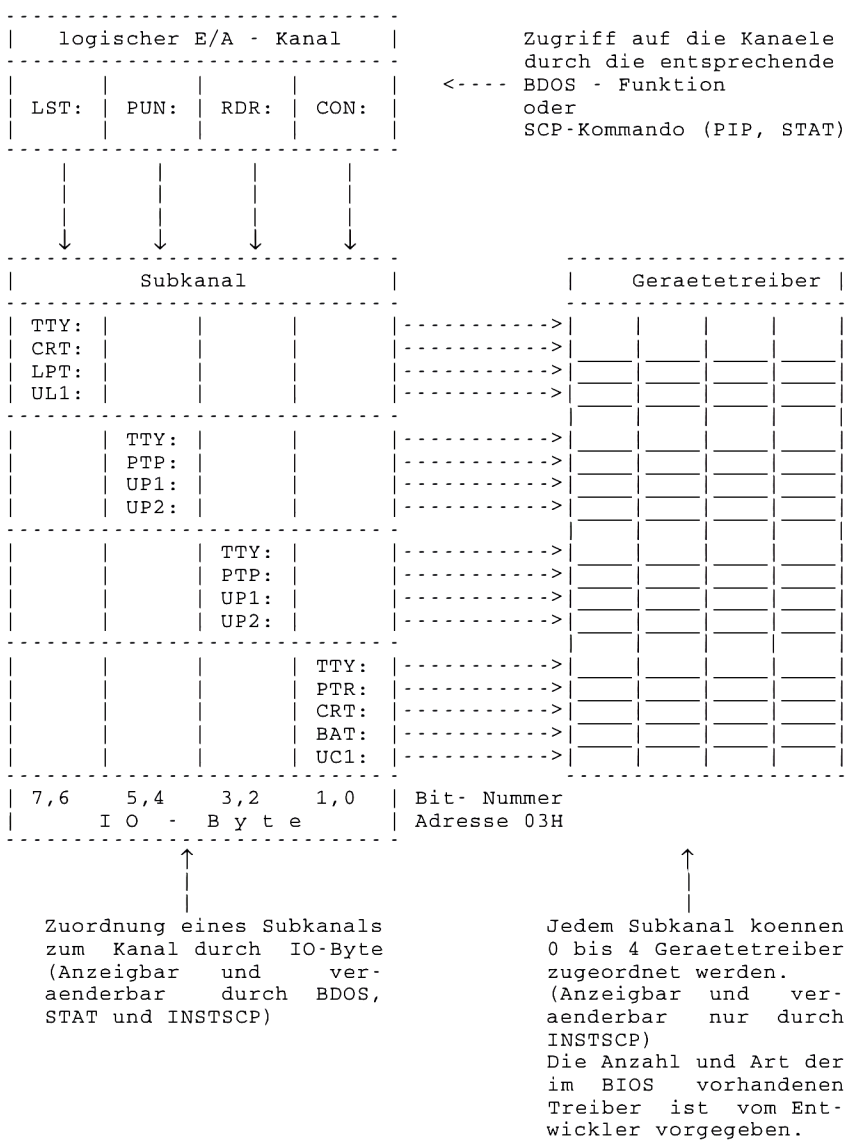
```
    +00 I    dr   I Laufwerkcodierung
        I         I dr = 0 : Bezugs-LW
        I         I      1 : LW A
        I         I     16 : LW P

    +01 I f1 - f8 I Dateiname in ASCII
        I         I Flags in den 8. Bits f1' - f8"
        I         I fuer Nutzerflags reserviert
    +09 I t1 - t3 I Dateityp in ASCII
        I         I Flags in den 8. Bits t11 - t3'
        I         I t1' = 1/0 : geschuetzte/ungeschuetzte Datei
        I         I t2' = 1/0 : SYSTEM-Verzeichnis-Datei
        I         I t3'       : reserviert
    +12 I    ex   I laufende Nummer des Dateiteils
        I         I (Extent) 0..31
    +13 I  s1 - 2 I reserviert fuer internen Gebrauch
    +15 I    rc   I Satz-zaehler innerhalb laufenden Extents
    +16 I d0 - 15 I Blockverzeichnis des Extents
    +32 I    cr   I laufende Satznummer der sequentiellen
        I         I Datei innerhalb des Extents
    +33 I  r0 — 2 I Satznummer der Datei fuer Direktzugriff
    +36 I         I Ende FCB
```

Aufbau der LW-Vektoren

```
Vektor im CPU-Doppelregister
bit 0 : LW A
    1 : LW B
    : : :: :
   15 : LW P
```

Uebersicht: Zuordnung KANAL-SUBKANAL-GERAETETREIBER



Der Anhang D ist nicht im Inhaltsverzeichnis aufgeführt und wurde vermutlich erst später in das Konzept eingefügt.

robotron

VEB Robotron
Büromaschinenwerk Sömmerda

DDR - 523 Sömmerda
Weißenseer Straße 52

robotron

Robotron Export-Import

Volkseigener
Außenhandelsbetrieb der
Deutschen Demokratischen
Republik
DDR - 108 Berlin
Friedrichstraße 61

R1 51-86 IV-27-1 0-3211